VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

DIPLOMOVÁ PRÁCE

Brno, 2021 Bc. Štefan Šuňal

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ
DEPARTMENT OF TELECOMMUNICATIONS

PŘIJÍMACÍ DEKODÉR RTTY
RECEIVING RTTY DECODER

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — **Bc. Štefan Šuňal**
AUTHOR

VEDOUCÍ PRÁCE — **doc. Ing. Ivo Lattenberg, Ph.D.**
SUPERVISOR

BRNO 2021

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Diplomová práce

magisterský navazující studijní program **Telekomunikační a informační technika**

Ústav telekomunikací

Student: Bc. Štefan Šuňal ***ID:*** 186213
Ročník: 2 ***Akademický rok:*** 2020/21

NÁZEV TÉMATU:

Přijímací dekodér RTTY

POKYNY PRO VYPRACOVÁNÍ:

S využitím mikrokontroléru ATMega či STM32 realizujte přijímací dekodér radiodálnopisného provozu na amatérských pásmech. Vstupem bude nízkofrekvenční signál z radiopřijímače. Zařízení by mělo být napájeno bateriově a mělo by fungovat autonomně. Bude vybaveno víceřádkovým displejem, kde se bude zobrazovat dekódovaný text a bude možno pomocí displeje konfigurovat nastavení detektoru. Při připojení k PC přes USB bude navíc možno pomocí programu vytvořeného v jazyce C# konfigurovat dekodér a zobrazovat přijatá data v této aplikaci.

DOPORUČENÁ LITERATURA:

[1] BRTNÍK, Bohumil a David MATOUŠEK. Mikroprocesorová technika: [práce s mikrokontroléry řady ATMEL AVR ATXmega A4]. Praha: BEN - technická literatura, 2011. ISBN 978-80-7300-406-4.

[2] MATOUŠEK, David. Práce s mikrokontroléry ATMEL. 2. vyd. Praha: BEN - technická literatura, 2006. μC & praxe. ISBN 80-7300-209-4.

Termín zadání: 1.2.2021 ***Termín odevzdání:*** 24.5.2021

Vedoucí práce: doc. Ing. Ivo Lattenberg, Ph.D.

prof. Ing. Jiří Mišurec, CSc.
předseda rady studijního programu

ABSTRAKT

Táto diplomová práca sa zaoberá návrhom dekodéru pre rádiovú komunikáciu RTTY, a jeho následnou implementáciou. Teoretická časť sa venuje bližšiemu popisu technológie RTTY a princípu fázovej modulácie FSK. Pozornosť je venovaná najmä jej domodulácii. Porovnáva vybrané dostupné platformy vhodné pre túto aplikáciu.
Praktická časť sa zaoberá návrhom a realizáciou plošného spoja. Následne popisuje implementáciu programu pre mikrokontrolér STM32 v jazyku C++. Súčasťou projektu je aj aplikácia na PC, ktorá je napísana v jazyku C# a jej rozhranieje implementované pomocou WPF.
Boli implementované 3 metódy softvérovej demodulácie, jedna za pomoci algoritmu FFT a dve na základe merania periódy obdĺžnikového signálu. Odolnosť týchto metód voči rušeniiu bola overená meraním.

KĽÚČOVÉ SLOVÁ

Arduino, Baudotov kód, C#, demodulácia, ďalekopis, FFT, FSK, RTTY, STM32

ABSTRACT

This thesis deals with designing and implementing autonomous decoder for RTTY communication. Theoretical part of the thesis deals with researching the RTTY technology and FSK modulation, which it uses, focusing mainly on the methods of demodulation. Next part of the thesis compares suitability of available platforms.
Practical part of the thesis describes the design and implementation of circuit board, firmware and computer application. The device is using STM32 microcontroller. Firmware was developed in C++ using Arduino. The computer application was created by using C# and WPF.
3 methods of FSK demodulations were implmented. One is using a FFT algorithm. Other two calculate frequency based on the period of square signal.

KEYWORDS

Arduino, Baudot code, C#, demodulation, FFT, FSK, RTTY, STM32, teletype

ŠUŇAL, Štefan. *Přijímací dekodér RTTY*. Brno, 2021, 76 s. Diplomová práca. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedúci práce: doc. Ing. Ivo Lattenberg, Ph.D.

VYHLÁSENIE

Vyhlasujem, že svoju diplomovú prácu na tému „Přijímací dekodér RTTY“ som vypracoval samostatne pod vedením vedúceho diplomovej práce, s využitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor uvedenej diplomovej práce ďalej vyhlasujem, že v súvislosti s vytvorením tejto diplomovej práce som neporušil autorské práva tretích osôb, najmä som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a/alebo majetkových a som si plne vedomý následkov porušenia ustanovenia § 11 a nasledujúcich autorského zákona Českej republiky č. 121/2000 Sb., o práve autorskom, o právach súvisiacich s právom autorským a o zmene niektorých zákonov (autorský zákon), v znení neskorších predpisov, vrátane možných trestnoprávnych dôsledkov vyplývajúcich z ustanovenia časti druhej, hlavy VI. diel 4 Trestného zákonníka Českej republiky č. 40/2009 Sb.

Brno ....

podpis autora

Obsah

Zoznam obrázkov

Zoznam tabuliek

Úvod

RTTY predstavuje jednu z prvých digitálnych komunikačných technológií. Táto technológia sa už prakticky nepouživa, ale je stále populárna medzi rádioamatérmi. Aj keď bola nahradená novšímy technológiami, stále predstavuje historický príklad použitia FSK modulácie. Množstvo princípov používaných v RTTY sa prenieslo do dnes používaných technológií. 5-bitové kódy, ktoré využíva sú predchodcom dnes populárneho ASCII kódovania. Štúdium týchto kódovaní objasňuje použitie niektorých znakov ASCII kódovania ako napríklad LF a CR, ktoré vyplívajú z použitia mechanických ďalekopisných prístrojov.

Cielom tejto práce je naštudovanie komunikácie RTTY a na základe nadobudnutých informácií následne navrhnúť zariadenie, ktoré umožňuhe RTTY dekódovať. Toto zariadenie by malo byť autonómne a umožňovať prepojenie s PC. Súčasťou zariadenia by mal byť aj displej zobrazujúci dekódovanú RTTY komunikáciu a tlačidlá pre nastavenie zariadenia. Dekodér by mal byť napájaný pomocou batérie, ktorá by mala byť nabíjatelná. Výsledkom projektu by malo byť funkčné zariadenie spolu s aplikáciou na počítač, ktorá umožňi jeho ďalšie nastavenie a zobrazenie dekódovaných dát.

V prvej kapitole práce je popísaná história ďalekopisu a technológie RTTY. Obsahuje tiež popis 5-bitových kódovaní, ktoré sú týmito technológiami používané a ich technické špecifikácie.

V druhej kapitole je popísané kľúčovanie frekvenčňym posunom spolu s jeho moduláciou a demoduláciou. Je tu uvedených aj niekoľko metód na demoduláciu takto modulovaného signálu.

V tretej kapitole sú popísané dostupné rady mikroprocesorov vhodné pre túto prácu. Sú tu popísané kritéria na ktorých bude voľba procesoru založená. Parametre procesorov sú porovnané pomocou tabuľky, na základe ktorej bol jeden zvolený.

V štvrtej kapitole je vysvetlený návrh zariadenia. Zahŕňa popis jednotlivých blokov a výpočty použité pri ich návrhu. Popisuje ďalej návrh a fyzickú realizáciu plošného spoja. Je tu tiež vysvetlený postup, použitý pri jeho osádzaní a sprevádzkovaní.

Piata kapitola sa venuje komunikačnému protokolu, ktorý bol vytvorený na komunikáciu medzi zariadením a aplikáciou. Sú tu popísané jednotlivé správy, ktoré medzi sa medzi zariadeniami posielajú. Vysvetluje taktiež spôsob naviazania a ukončenia komunikácie.

Kapitola čislo šesť sa zaoberá programovou realizáciou softwéru pre mikrokontrolér. Na jej začiatku je popísaný dôvod výberu konkrétnej prgramvej platformy. Ďalej obsahuje popis algoritmov použitých pri demodulácii a dekódovaní signálu. Popisuje tiež užívateľské rozhranie zariadenia.

V siedmej kapitole je popísaná aplikácia na PC. Kapitola sa najskôr venuje popisu platformy .NET a jazyka C#. Následne je v nej popísané fungovanie programu a užívateľské rozhranie aplikácie.

Témou ôsmej kapitoly bolo navrhnutie postupu ako overiť funkčnosť zariadenia na zarušenom signále. Je tu popísaný postup merania, použitý softvér a hárdvér. Obsahuje aj merané výsledky spracované do tabulky a porovnanie jednotlivých metód.

1 RTTY

Radio Teletype, v preklade rádiový ďalekopis, predstavuje technológiu umožňujúcu prenos správ medzi ďalekopisnými prístrojomi pomocou rádia.

1.1 História

Ďalekopisná komunikácia bola prvýkrát použitá v roku 1849 na linke Philadelphia-New York. Z dôvodu mechanickej zložitosti a najmä vysokej ceny prístrojov, sa tento systém neujal. Jednou z prekážok pre efektívnu ďalekopisnú komunikáciu bola forma v akej boli dáta prenášané. Štandardom pre diaľkovú komunikáciu bol v tejto dobe Morseov kód prenášaný pomocou telegrafov. Vlastnosti, vďaka ktorým bol tento kód vhodný pre telegrafnú komunikáciu a robili ho ľahko dekódovatelným pre zaškoleného operátora, ako rôzna dĺžka znakov a použitie dlhých a krátkych signálov, komplikovali jeho strojové dekódovanie. V roku 1874 vytvoril francúzsky inžinier Émile Baudot kód, ktorý dal základ modernej digitálnej komunikácii. Baudotov kód využíval namiesto dlhých a krátkych signálových prvkov, dva rovnako dlhé signálové prvky zodpovedajúce logickej jednotke a nule. Každému znaku bol pritom priradený kód rovnakej dĺžky 5 bitov. Spolu s týmto kódom si Emilé Baudot patentoval aj návrh ďalekopisného systému pre jeho praktické použitie. Na generovanie kódu v ňom bola použitá jednoduchá klávesnica s piatimi klávesami, kde každá klávesa reprezentovala jeden bit kódu. Na potreby prijímania a dekódovania signálu nahradila operátora jednoduchá tlačiareň, ktorá tlačila prijaté znaky na pásku [1].

Tento systém si našiel okamžité uplatnenie vo Francúzsku a počas nasledujúcich desaťročí bol postupne vylepšovaný. Prístroje schopné generovať znaky na základe dierových štítkov postupne nahradili klávesnice. Rýchlosť vysielania teda už nebola závislá na ľudskom faktore, ale len na rýchlosti jednotlivých častí systému. Experimentovalo sa tiež s prenosom signálu pomocou rádiových vĺn. Prvé komerčné ďalekopisné systémy využívajúce rádio vznikli v medzivojnovom období v USA. Pre svoj prenos využívali frekvenčnú moduláciu spolu s upravenou verziou Baudotovho kódu. Počas 2. svetovej vojny našli tieto systémy aj široké armádne použitie. Mimo bežnej komunikácie, prebehli aj pokusy na komunikáciu s posádkami tankov alebo lietadiel [1] [2].

Po skončení vojny sa prebytočné a zastaralé armádne zásoby, podporujúce RTTY, začali dostávať do rúk verejnosti. Jej dostupnosť a jednoduchosť používania ju spopularizovala v rastúcej komunite rádioamatérov. Až do 70. rokov 20. storočia predstavovala RTTY jednu z najrozšírenejších metód rádiovej komunikácie. Pôvodné elektromechanické ďalekopisné prístroje postupne začali byť nahrádzované počítačmi, na ktorých bežal RTTY emulátor. Táto technológia je stále v určitej

miere využívaná v námornícte a armáde. Existujú tiež niektoré vysielania ponúkajúce predpovede počasia. Väčšina vysielaní je však v dnešnej dobe realizovaná v rámci rádioamatérskych súťaží [3].

1.2 5-bitové kódovania

Prenos RTTY je štandardne kódovaný pomocou 5-bitových kódov. Historicky sa používalo viacero variácií. Najdôležitejšími sú však Baudotov kód, označovaný tiež ITA-1, a na ňom založený kód ITA-2.

Abeceda baudotovho kódu zobrazená v tabulke bola zadávaná na klávesnici kde bity v stĺpcoch I, II a III boli zadávané pravou rukou a bity IV a V ľavou rukou operátora. Kód je preto prispôsobený tak, aby boli pohyby rozdelené rovnomerne medzi obe ruky operátora a predišlo sa tak jeho únave. Tento kód bol relatívne jednoduchý a neumožňoval detekciu, ani opravu chýb, a je preto citlivý na rušenie [3] [4].

V-IV	Písmená	Symboly	I-II-III
0-0			0-0-0
0-0	A	1	1-0-0
0-0	É	&	1-1-0
0-0	E	2	0-1-0
0-0	I	o	0-1-1
0-0	O	5	1-1-1
0-0	U	4	1-0-1
0-0	Y	3	0-0-1
0-1	B	8	0-0-1
0-1	C	9	1-0-1
0-1	D	0	1-1-1
0-1	F	f	0-1-1
0-1	G	7	0-1-0
0-1	H	h	1-1-0
0-1	J	6	1-0-0
0-1	*F*	*Blank*	0-0-0

V-IV	Písmená	Symboly	I-II-III
1-1	*Mazanie*	*Mazanie*	0-0-0
1-1	K	(	1-0-0
1-1	L	=	1-1-0
1-1	M	)	0-1-0
1-1	N	N°	0-1-1
1-1	P	%	1-1-1
1-1	Q	/	1-0-1
1-1	R	–	0-0-1
1-0	S	;	0-0-1
1-0	T	!	1-0-1
1-0	V	’	1-1-1
1-0	W	?	0-1-1
1-0	X	,	0-1-0
1-0	Z	:	1-1-0
1-0	t	.	1-0-0
1-0	*Blank*	*Letter*	0-0-0

Tab. 1.1: Abeceda ITA-1 [4]

S príchodom modernejších ďalekopisných pristrojov, založených na písacom stroji bolo potrebné tento kód upraviť. Nová abeceda dbala viac na to, koľko mechanických operácií je potrebných pre vyslanie, prijatie a zapísanie znaku v závislosti na

frekvencii jeho výskytu. Kód bol tiež rošírený o riadiace znaky Carriage Return a Line Feed. Tento kód bol v roku 1924 štandardizovaný ako Mediznárodná telegrafná abeceda 2 (ITA-2). Rovnako ako ITA-1 neumožňuje detekciu ani opravu chýb. Pri RTTY komunikácii je toto kódovanie obvykle kombinované so start a stop bitmy, aby komunikácia mohla prebiehať synchrónne.

Kód	Písmená	Symboly
000-11	A	-
110-01	B	?
011-10	C	:
010-01	D	$
000-01	E	3
011-01	F	!
110-10	G	&
101-00	H	Stop
001-10	I	8
010-11	J	’
011-11	K	(
100-10	L	)
111-00	M	.
011-00	N	,
110-00	O	9
101-10	P	0

Kód	Písmená	Symboly
101-11	Q	1
010-10	R	4
001-01	S	BEL
100-00	T	5
001-11	U	7
111-10	V	;
100-11	W	2
111-01	X	/
101-01	Y	6
100-01	Z	"
000-00	Blank	
010-00	CR	
000-10	LF	
001-00	SP	
111-11	LS	
110-11	FS	

Tab. 1.2: Abeceda ITA-2 [4]

Znaky týchto kódov sú reprezentované piatimi bitmi. To znamená, že je pomocou nich možné zakódovať len 32 rôznych znakov. V bežnej písanej komunikácii sa ich však obvykle vyskytuje viac. Z tohoto dôvodu bola väčšine kódov priradená dvojica znakov, jedno písmeno a jeden symbol. Dva z kódov boli vyhradené pre ako riadiace znaky prepínanie medzi módom písmen a symbolov. To ako bude daný kód dekódovaný je určené podľa toho, ktorý z týchto dvoch riadiacich znakov bol prijatý naposledy [3].

Štandardná abeceda ITA-2 obsahuje 6 kontrolných znakov. Znak CR posunie kurzor, alebo v prípade mechanického prístroja valec s papierom, na začiatok riadku. Prijatím znaku LF indikovalo prístroju, že má posunúť valec s papierom o riadok nižšie. Pri novších systémoch je ale tento znak často používaný k posunu na začiatok nového riadku. Znak SP predstavuje medzeru v texte. Po prijatí znaku BEL obvykle zazvonil zvonček, ktorý informoval operátora o prichádajúcom prenose. Znak Blank

predstavuje prázdny znak alebo prázdnu pásku. Pri dekódovaní pomocou počítača je zameniteľný so znakom NULL. Kontrolné znaky FS a LS zabezpečujú prepínanie medzi módom prijímania písmen a čísel [4].

Keďže sú tieto kódovania náchylné na vznik chýb, môže dôjsť k strate symbolu FS alebo LS. Tým by mohlo dôjsť k znehodnoteniu ďalších všetkých dát do prijatia ďalšieho riadiaceho symbolu. Z tohoto dôvodu je obvykle považovaný za predvolený mód LS. V prípade odosielania symbolov sa odošle znak FS a určitý počet po ňom prijatých znakov považujeme za symboly. V niektorých prípadoch je znak FS posielaný pred každou päticou bitov reprezentujúcou symbol [3].

1.3 Technické špecifikácie

RTTY systém je obvykle založený na použití kódu ITA-2, experimentovalo sa aj s použitím kódovania ASCII, ktoré sa ale nedostalo do bežného užívania. Komunikácia je asynchrónna, s využitím 1 start bitu a 1, 1,5 alebo 2 stop bitov. Start bit obvykle zodpovedá úrovni logickej 1 a stop bity majú úroveň logickej 0. Prenos signálu je založený na binárnom frekvenčno kľúčovaní. Frekvenčné posuny majú obvykle hodnoty 85 hz, 170 Hz, 425 Hz, 450 Hz alebo 850 Hz. Bežné prenosové rýchlosti sú 45, 50, 75, 100,150 a 300 baudov. Komerčné ďalekopisné systémy fungovali obvykle na rýchlosti 50 bd s posunom 425 alebo 850 Hz. Najpoužívanejšou bola však rýchlosť 45 bd v kombinácií s posunom 170 Hz, ktorá je štandardne používaná rádioamatérmi. Využívalo sa viacero frekvenčných párov, pre rádioamatérsky prenos sa ale zaužívala frekvencia 2125 Hz pre značku a 2195 Hz pre medzeru [3].

2 Frekvenčné klúčovanie (FSK)

Frekvenčné klúčovanie je technika frekvenčnej modulácie založená na diskrétnych zmenách nosnej vlny. Najjednoduchšia forma FSK sa označuje ako binárne FSK a využíva dva rôzne signály s frekvenciami f_1 a f_2. Pre frekvencie sa používa označenie Mark a Space, pričom Mark označuje frekvenciu zodpovedajúcu logickej 1 a space frekvenciu zodpovedajúcu logickej 0. Trvanie jedného signáloveho prvku mark alebo space v sekundách môžme označiť T. Medzi frekvenciami f_1 a f_2 sa nachádza menovitá stredová frekvencia f_c, ktorú určíme ako

$$f_c = \frac{f_1 + f_2}{2} \tag{2.1}$$

a platí pre ňu

$$f_1 = f_c + \Delta f, \tag{2.2}$$

$$f_2 = f_c - \Delta f, \tag{2.3}$$

kde Δf predstavuje frekvenčnú deviáciu. Vzdialenosť medzi frekvenciami f_1 a f_2 nazývame frekvenčný posun alebo shift [5].

2.1 Modulácia FSK

2.1.1 Koherentná modulácia

Ak má fáza každého symbolu space a mark pevný vzťah vzhľadom na referenciu, hovoríme o koherentnom signále. Podľa [5] ho je možné popísať vzorcami

$$S_1(t) = A\cos(2\pi f_1 t + \phi) \text{ pre } 1, \tag{2.4}$$

$$S_2(t) = A\cos(2\pi f_2 t + \phi), \text{ pre } 0. \tag{2.5}$$

Vzniká napríklad v prípade, že je na generovanie oboch signálov použitý jeden oscilátor s možnosťou prepínania frekvencie[6].

2.1.2 Nekoherentná modulácia

V prípade, že sú fázy signálov nezávislé môže byť táto demodulácia označená ako nekoherentná. Signály v tom prípade môžme popísať vzorcami

$$S_1(t) = A\cos(2\pi f_1 t + \phi_1) \text{ pre } 1, \tag{2.6}$$

$$S_2(t) = A\cos(2\pi f_2 t + \phi_2), \text{ pre } 0, \tag{2.7}$$

kde ϕ_1 a ϕ_2 predstavujú počiatočnú fázu signálov [5].

K nekoherentnej modulácií dochádza, ak na generovanie frekvencií používame dva nezávisilé oscilátory. Fáza výsledného FSK signálu sa pri zmene frekvencie bude meniť skokovo [6].

2.2 Demodulácia FSK

Rovnako ako pri modulácii je možné signál demodulovať koherentne alebo nekoherentne. Koherentný signál je pritom možné demodulovať koherentne aj nekoherentne, zatiaľ čo nekoherentný signál len nekoherentne. Pre demoduláciu koherentného signálu sa často volí možnosť použitia nekoherentného demodulátora, keďže si nevyžaduje rekonštrukciu nosnej frekvencie. Koherentnosť signálu FSK, ktorý budeme spracovávať, nie je garantovaná. Bude preto kladený dôraz najmä na metódy nekoherentnej demodulácie [5].

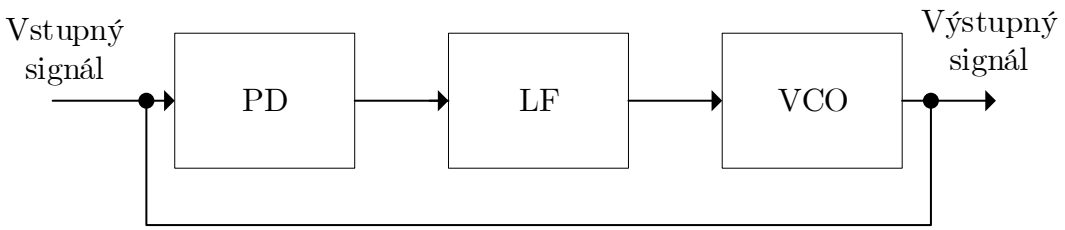
Demodulácia štandardne pozostáva z predspracovania signálu obvykle zahŕňajúceho odstránenia jednosmernej zložky a prefiltrovanie signálu pásmovou priepusťou tak, aby v ňom ostali len pre nás dôležité frekvencie. V takto upravenom signále je následne potrebné detekovať frekvencie zodpovedajúce mark a space. Podla typu demodulácie môže byť detekcia realizovaná jedným detektorom, ktorý dokáže určiť obe frekvencie alebo pomocou dvoch separátnych detektorov, ktorých výstup je následne vyhodnotený porovnávacim členom. Detekciu je možné realizovať softwarovo, alebo za použitia elektrických obvodov. Existuje množstvo zaužívaných integrovaných obvodov a algoritmov, ktoré sa však líšia v náročnosti ich aplikácie a v odolnosti voči rušeniu a chybám. Aby sme mohli signál ďalej digitálne spracovávať a dekódovať je ho potrebné dostať do digitálnej podoby. Časť demodulácie, v ktorej je signál digitalizovaný, závisí od použitej alternatívy demodulátora. Pri jeho digitalizácii je pre predídenie aliasingu potrebné dodržať Nyquistov vzorkovací teorém

$$f_s \geq 2f, \tag{2.8}$$

kde f_s je vzorkovacia frekvencia a f je maximálna frekvencia signálu [5].

2.2.1 Fázový záves

Jednou z možností demodulácie FSK je použitie Fázového závesu (PLL). PLL je často používané v integrovaných obvodoch pre demoduláciu FM signálov alebo pre tónovú voľbu. Ako je možné vidieť na obrázku 2.1, pozostáva z troch blokov. Fázový detektor (PD) predstavuje časť obvodu, ktorá porovnáva vstupné signály, a na základe ich fázového rozdielu generuje výstupné napätie. Filter slučky(LF) filtruje z výstupu PD nechcené časti vstupného signálu. Oscilátor riadený napätím (VCO) generuje signál, ktorý je považovaný za výstup. Ak na PLL obvod neprivedieme žiaden



Obr. 2.1: Bloková schéma fázového závesu

vstupný signál, na vstupe PD je iba referenčný signál generovaný VCO. Na vstupe VCO teda bude privedené určité jednosmerné napätie. Toto napätie je konštanté a zodpovedá frekvencii označovanej ako vlastná frekvencia. Ak na vstup PLL bude privedený signál, bude jeho napätie a fáza porovnávaná so referenčným signálom. Na výstupe PD vzniká napätie závislé na rozdiele jeho vstupných signálov. Toto napätie je privedené na vstup VCO, ktorého frekvencia a fáza sú preladené na totožné so vstupným signálom [7].

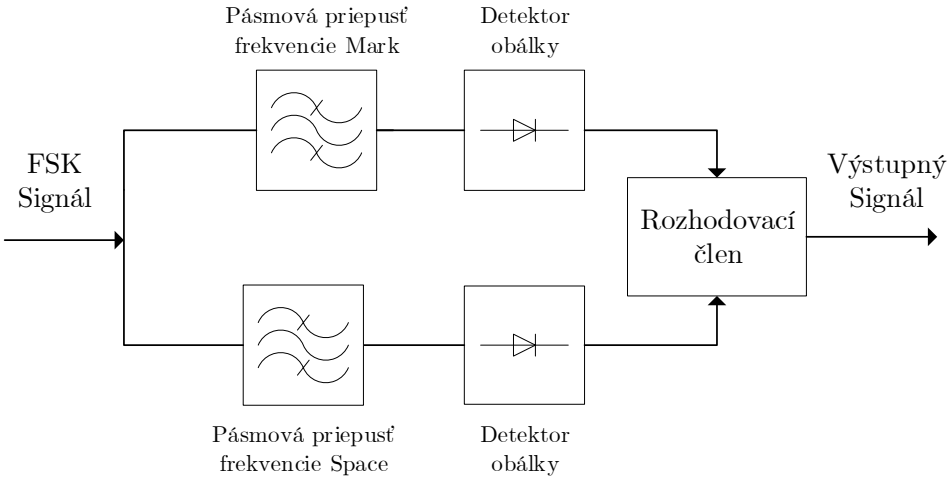
Vlastnosti PLL sú definované pomocou dvoch oblastí. Oblasť udržania (lock-in range) vyjadruje v akom rozsahu frekvencie sa je schopný systém udržať v závese. Rozsah, v ktorom sa je schopný systém dostať do závesu je vždy menší ako oblasť udržania a je označovaný rozsah zachytenia (hold-in range) [7].

Jedným z možných použití PLL je demodulácia FM signálu. Pri privedení fázovo modulovaného signálu na vstup PLL bude VCO sledovať jeho frekvenciu. Podľa veľkosti napätia privádzaného na vstup VCO je potom možné určiť hodnotu frekvencie [7].

2.2.2 Demodulácia pomocou filtrov

Princíp týchto demodulátorov je založený na rozdelení tohoto signálu na dve zložky obsahujúce frekvencie mark alebo space. Vhodnou úpravou a porovnaním týchto zložiek je potom možné určiť, ktorá z týchto frekvencií sa v určitom čase v signále nachádza [5].

Na realizáciu tohoto demodulátoru je možné použiť dve paralelné pásmové priepuste spolu s detektormi obálky. Toto riešenie je zobrazené na obrázku 2.2. Jedna pásmová priepusť je určená pre frekvenciu mark a druhá pre frekvenciu space. Výstupom týchto filtrov bude signál obsahujúci už len príslušnú frekvenciu. Aby bolo možné tieto signály porovnať je najprv detekovaná ich obálka. Výsledná frekvencia je určená porovnaním obálok týchto signálov [5].



Obr. 2.2: Bloková schéma demodulátora za použitia filtrov

Pri návrhu filtrov pre túto metódu je potrebné vziať do úvahy parametre signálu a predpokladaného rušenia. Jeho hlavnou nevýhodou je potreba návrhu a nastavenia filtrov. V prípade hardwérovo realizovaných filtrov je toto nastavenie realizované pomocou RC členov. Prepínanie medzi nastaveniami filtra by bolo teda relatívne zložité. Softwérová realizácia je o niečo flexibilnejšia, ale pre efektívnu implementáciu si rovnako vyžaduje návrh a výpočet konštánt filtra .

2.2.3 Detekcia prechodu nulou

Hodnota napätia striedavého signálu reprezentovaného funkciou sínus sa v čase periodicky mení a prechádza medzi zápornými a kladnými hodnotami. Táto metóda je založená na počítaní týchto prechodov a určovaní frekvencie na základe ich množstva alebo časového rozdielu. Jedna perióda signálu zodpovedá trom prechodom nulou. Časový rozdiel medzi prvým a posledným z týchto prechodov teda zodpovedá perióde signálu T, z ktorej je možné určiť frekvenciu f

$$f = \frac{1}{T}. \tag{2.9}$$

Závislosť tejto metódy na tvare signálu ju robí citlivu voči rušeniu. Je potrebné preto signál vhodne predspracovať pomocou dolnej alebo pásmovej priepuste. Pre zvýšenie presnosti je tiež vhodné aplikovať túto metódu na viacero po sebe nasledujúcich periód signálu [8].

2.2.4 Fourierova transformácia

Fourierovu transformácia X(f) umožňuje prevod signálu z časovej do frekvenčnej oblasti a je ju možné vyjadriť ako

$$X(f) = \int_{-\infty}^{+\infty} x(t)e^{-j2\pi ft}dt \tag{2.10}$$

kde $x(t)$ je spojitý signál v časovej oblasti. Analýzou výrazu X(f) dokážeme určiť frekvenčnú skladbu daného signálu. Fourierova transformácia teda môže byť použitá ako detektor frekvencií pri demodulácii FSK.

Pre spracovanie digitálneho signálu sa využíva diskrétna fourierova transformácia (DFT), ktorej rovnica má tvar

$$X(m) = \sum_{n=0}^{N-1} x(n)e^{\frac{-j2\pi nm}{N}} = \sum_{n=0}^{N-1} x(n)\cos\left(\frac{2\pi nm}{N}\right) - j\sum_{n=0}^{N-1} x(n)\sin\left(\frac{2\pi nm}{N}\right). \tag{2.11}$$

kde $x(n)$ predstavuje diskrétnu sekvenciu vzoriek získaných zo spojitej premennej $x(t)$ a N je veľkosť transformácie.

Rýchla Fourierova transformácia

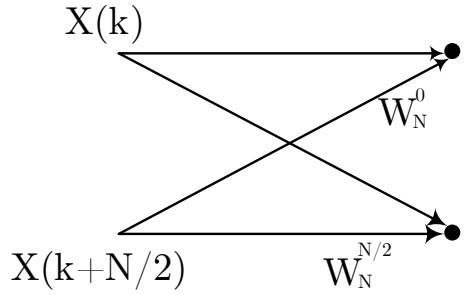
Výpočet štandardnej DFT si vyžaduje spraviť N^2 komplexných súčinov, čo ho robí veľmi časovo náročným. Pre zefektívnenie vzniklo množstvo algoritmov ktoré sa súrhnne nazývajú Rýchla Fourierova transformácia (FFT). Využívajú to, že spektrum signálu s párnym počtom prvkov je syemtrické a komplexne združené okolo stredu, a výpočet DFT je teda možné rozdeliť na párne a nepárne zložky, pričom dĺžka každej z nich bude polovičná oproti pôvodnej DFT. Ak počet vzoriek vstupného signálu zodpovedá 2^M, je možné tieto DFT ďalej rozdeľovať až dokým nevzniknú DFT s veľkosťou 2. Tie sú často zobrazované pomocou takzvaných motýlikových diagramov [9]. Z výpočtu je taktiež možné vyjadriť exponenciálny člen, označovaný tiež ako twiddle factor, ktorý je nezávislý od poradia vstupnej vzorky a môžeme ho zapísať ako

$$W_N^m = e^{\frac{-j2\pi+m}{N}}. \tag{2.12}$$

Výpočtom získame spektrum signálu, ktoré môžeme ďalej analyzovať [9]. Frekvencie vyskytujúce sa v signále sa prejavia ako špic v spektre, a môžeme ich teda nájsť použitím vhodného algoritmu na detekciu lokálnych maxím.

Aby bolo možné FFT použiť na detekciu konkrétnej frekvencie je potrebné zohľadniť rozlíšenie získaného spektra. To sa bude skladať z frekvenčných čiar, označovaných po anglicky tiež ako bins (koše), ktorých počet vypočítame ako

$$\frac{N}{2} \tag{2.13}$$



Obr. 2.3: Motýlikový diagram 1. rádu

kde N predstavuje počet bodov alebo dĺžku spracovaného signálu. Prvá frekvenčná čiara, zodpovedajúca jednosmernej zložke, pritom bude ležať na frekvencii 0 Hz a posledná na frekvencii, ktorú získame ako

$$\frac{f_s}{2} - \frac{f_s}{N}. \tag{2.14}$$

kde f_s predstavuje vzorkovaciu frekvenciu spracovaného signálu. vzdialenosť týchto frekvenčńých liniek, vypočítame ako

$$\delta f = \frac{F_S}{N}. \tag{2.15}$$

Z týchto vzorcov vyplýva, že rozlíšenie FFT je závislé na zvolenej dĺžke FFT a že s množstvom vzorkov rastie presnosť výsledku, čo je ale spojené so zvýšením náročnosti výpočtu [10].

Algoritmus FFT je použiteľný ako detektor v demodulátore FSK. Demoduláciu je možné realizovať rozdelením navzorkovaného signálu na bloky zodpovedajúce jednému prenášanému symbolu. Aplikovanie FFT na jednotlivé bloky a následná analýza získaného spektra dovoluje určiť, ktorá z frekvencií mark alebo space sa v danom bloku nachádza. Bloky obsahujúce frekvenciu mark sú potom považované za logické jednotky a bloky s frekvenciou space za logické nuly. Aby sa predišlo chybám vznikajúcim pri prenose alebo generovaní signálu je vhodné, aby bol každý signálový prvok rozdelený na viac ako jeden blok [10].

Výhodou tohoto spôsobu demodulácie je možnosť jeho aplikácie na ľubovolné dve frekvencie, ktoré nieje potrebné fixne určiť hardwérovými prvkami. Je však obmedzený parametrami použitého procesora.

2.2.5 Goertzelov algoritmus

Nevýhodou FFT je, že aj v prípade potreby zistiť či signál obsahuje konkrétnu frekvenciu, je potrebné vypočítať celé jeho spektrum. Pre tento účel existuje rýchlejšia a efektívnejšia metóda v podobe Goertzelovho algoritmu. Ten umožňuje na základe navzorkovaného signálu vypočítať hodnotu len pre konkrétnu spektrálnu čiaru. V prípade základného Geortzelovho algortimu ako výsledok získame reálnu a imaginárnu časť ako pri bežnej DFT, z ktorých dokážeme ďalej určiť aj fázu a magnitúdu danej frekvencie. Pre Geortzelov algoritmus platia rovnaké pravidlá týkajúce sa rozlíšenia, počtu vzorkov a vzorkovacej frekvencie ako pre FFT. Narozdiel od FFT však počet vzorkov N nemusí byť mocninou dvojky [11].

Goertzelov algoritmus pre svoj výpočet používa niekoľko konštánt ktoré je potrebné vypočítať na základe nami zvolených parametrov f_s, N a hľadanej frekvencie f_h. Postup ich výpočtu je zobrazný v 2.16.

$$k \doteq 0,5 + \frac{N f_h}{f_s} \tag{2.16}$$

$$w = \frac{2\pi k}{N} \tag{2.17}$$

$$C = 2 * \cos(w) \tag{2.18}$$

Na výpočet sú ďalej potrebné tri premenné, ktoré budeme označovať Q_0,Q_1 a Q_2. Q_1 zodpovedá stavu Q_0 v predchádzajúcom výpočte a Q_2 zodpovedá stavu Q_0 z výpočtu pred predchádzajúcim. Q_1 a Q_2 musia byť na začiatku každéh ospracovávaného bloku vzoriek nastavené na hodnotu 0. Následne je potrebné spraviť nasledujúcu sériu výpočtov pre každý vzorok v bloku

$$Q_0 = C * Q_1 - Q_2 + vzorka \tag{2.19}$$

$$Q_2 = Q_1 \tag{2.20}$$

$$Q_1 = Q_2 \tag{2.21}$$

Keď bude tento výpočet vykonaný N-krát je možné pomocou týchto premenných získať reálnu a komplexnú časť signálu.

$$reálna = (Q_1 - Q_2 \cos w) \tag{2.22}$$

$$komplexná = (Q_2 \sin w) \tag{2.23}$$

Pre zrýchlenie je možné zanedbať výpočet reálnej a imaginárnej zložky a vypočítať priamo magnitúdu danej frekvencie. V tomto prípade by výpočet koeficientov Q prebehol rovnako ako v základnom algortime, ale výpočet reálnej a komplexnej zložky nahradíme výpočtom

$$magnitúda^2 = Q_1^2 + Q_2^2 - Q_1 Q_2 C. \tag{2.24}$$

Analýzou dát o magnitúde je následne možné určiť, napríklad porovnávaním s určitou prahovou hodnotou, či sa v signále daná frekvencia nachádza. Dvomi aplikáciami tohoto algoritmu je teda detekovatelné, či sa v časti signálu nachádza frekvencia mark alebo space [11].

3 Výber platformy

Pre realizáciu zariadenia vo forme, ktorá umožňuje ďalšie nastavenie a autonómnosť je potrebné v zariadení použiť mikrokontrolér. Tento kontrolér bude mať na starosť komunikáciu s PC pomocou rozhrania USB, zobrazovanie dát na displeji, demoduláciu a dekódovanie FSK signálu. Pri voľbe mikrokontroléru bude teda kladený dôraz najmä na:

- AD prevodník - Procesor bude spracovávať signál s frekvenciami medzi 1 a 3 kHz. AD prevodník musí teda zvládať vzorkovanie aspoň 6 kHz aby sa predišlo aliasingu. Ďalšie ADC vstupy nie sú nutné ale v prípade ich dostupnosti môže byť jeden využitý na meranie napätia batérie.
- Pre komunikáciu s aplikáciou v PC je nutné aby mikrokontrolér disponoval komunikačnou linkou. Najvhodnejšou variantou je rozhranie USB. Bolo by vhodné aby bolo procesorom priamo podporované. V opačnom prípade bude potrebné použiť prevodník USB na sériovú linku, napríklad SCI. Tento prevodník by však musel byť implementovaný hardvérovo.
- Na komunikáciu s displejom je možné použiť 6 pinov kontroléru alebo zbernicu I2C, ktorá si vyžaduje 2 piny. Použitím kontroléru využívajúceho túto zbernicu by sa zjednodušil návrh plošného spoja a znížili by sa nároky na počet pinov kontroléru.

Na trhu je dostupné množstvo rád kontrolérov od rôznych výrobcov. Z hladiska dostupnosti a mojich predchádzajúcich skúseností s programovaním, boli pre tento projekt zvažované mikrokontroléry z rodín ATmega a STM32.

ATmega

Séria kontrolérov ATmega je vyrábaná spoločnosťou Atmel. Tieto kontroléry využívajú technológiu AVR, ktorá predstavuje modifikovanú verziu harvardskej architektúry. Procesory v tejto sérii sú 8 bitové. Táto séria kontrolérov je určená na zložitejšie aplikácie. Má rozšírený inštrukčný set a funkcie oproti sérii ATtiny. Väčšina procesorov v tejto sérii taktiež podporuje debugging pomocou rozhrania JTAG. Kontroléry v rámci tejto série sa líšia veľkosťou pamätí Flash, SDRAM a EEPROM. Veľkosť flash pamäte je štandarnde zahrnutá v názve čipu a môže dosahovať až 256 kB. Odlišujú sa tiež tým, aké interné obvody obsahujú a aké funkcie podporujú. Ide napríklad o množstvo ADC prevodníkov, počet a rozlíšenie čitačov a časovačov alebo UART. Procesory ATmega nepoužívajú rozhranie I2C. Disponujú však rozhraním TWI, ktoré je kompatibilné s I2C Informácie o tom aké prvky daný procesor obsahuje sú obsiahnuté v jeho dokumentácii. Žiaden z procesorov tejto série neumožňuje použitie DMA, ktoré je dostupné až vo vyššej sérii ATxmega.

STM32

Kontroléry z rodiny STM32 sú vyrábané firmou STMicroelectonics. Sú založené na 32 bitovej ARM architektúre. Väčšina STM32 procesorov umožňuje použitie funkcie DMA, ktorá dokáže realizovať dátové prenosy medzi perifériami a pamäťou bez potreby priamej účasti procesoru. Umožňujú viacero možností debuggu pomocou SWD, JTAG alebo ST-Link. Kontroléry v tejto rodine sú rozdelené do niekoľkých sérií s rozličnými zameraniami. Základom týchto kontrolérov je jadro Arm Cortex-M, ktoré je vyrábané v rôznych verziách. Je pritom kladený dôraz na kompatibilitu medzi jednotlivými verziami. Výhodou pri vývoji pre tieto kontroléry je dostupnosť oficiálneho grafického nástroja, umožňujúceho nastavenie mikrokontroléru.

3.1 Porovnanie procesorov

	ATmega644PA	ATmega2560	STM32f103c8	STM32f411CE
Maximálna Taktovacia Frekvencia	16 MHz	16 MHz	72 MHz	100 MHz
Programová pamäť	64 kB	256 kB	64 kB	256 kB
SRAM	20	20		128 kB
Napájacie napätie	2,7 V-5,5 V	1,8 V-5,5 V	2,0 V-3,6 V	1,7 V-3,6 V
Rozlíšenie ADC	10 b	10 b	12 b	12 b
I2C	1	1	2	3
UART	2	4	3	3
USB	Nie	Nie	Áno	Áno
DMA	Nie	Nie	Áno	Áno

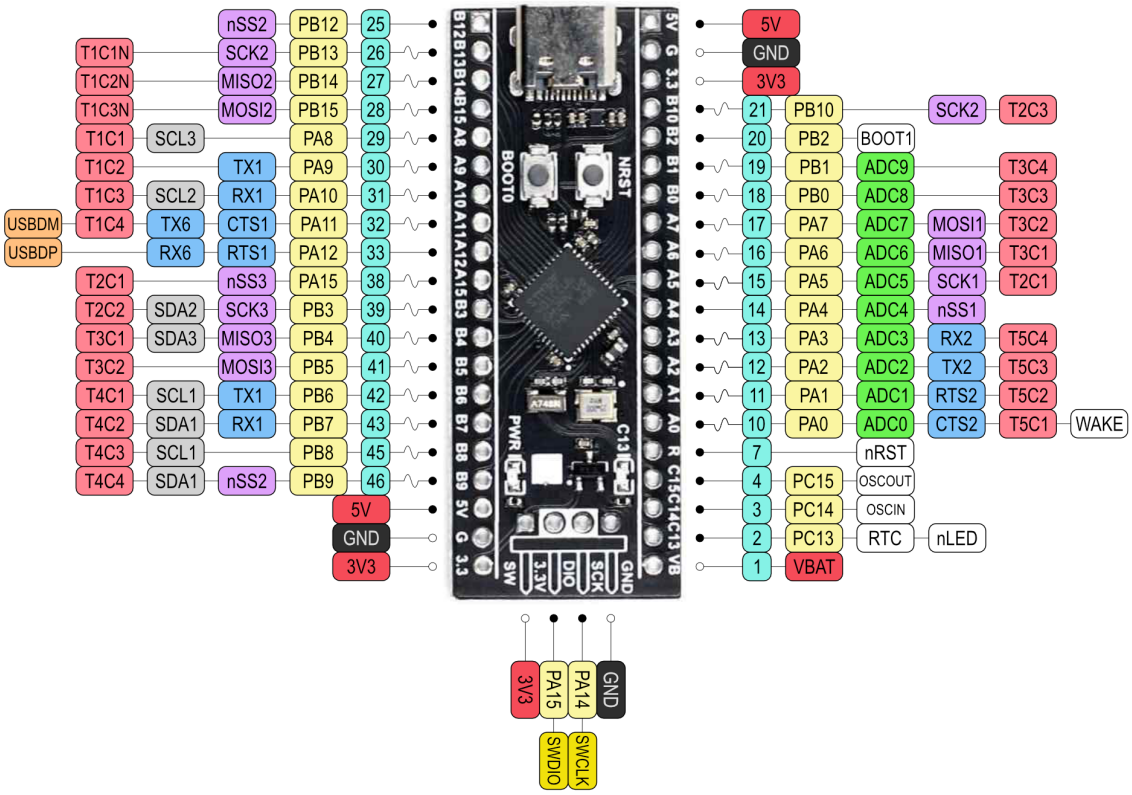
Tab. 3.1: Parametre porovnávaných mikrokontrolérov[12][13][14][15]

Údaje o procesoroch zapísané v tabulke 3.1 boli získané z ich príslušnej dokumentácie. Na základe týchto parametrov bolo rozhodnuté o použití procesoru z rodiny STM32. Ponúkajú vyššie taktovacie rýchlosti a teda aj rýchlosti ADC, ako aj podporu komunikačných rozhraní USB. Výhodou je aj dostupnosť DMA v týchto procesoroch, pomocou ktorého je možné zautomatizovať prenos dát medzi perifériami a pamäťou, čím by bolo možné efektívnejšie využívať dostupné prostriedky procesora.

3.2 Blackpill

Pre prvú verziu plošného spoja bola zvolená varianta využívajúca dosku Bluepill. Tá je osadená procesorom STM32F103C8T6. Doska zahŕňa aj obvody zabezpečujúce prevod napájacieho napätia 3,3 V vhodného pre procesor. Súčasťou dosky sú tiež vyvedené piny pre debugging a programátor. Bol zvolený najmä z dôvodu dostupnosti a zjednodušenia montáže, keďže osadenie SMD prevedenia procesoru by bolo v domácom prostredí zložité. Výhodou je tiež možnosť opakovaného použitia a nízka cena [16].

V najnovšej verzii bola nahradená doskou Blackpill, ktorá využíva výkonnejší mikrokontrolér STM32f411CE. Podobne ako doska Bluepill, obsahuje obvody pre úpravu napájacieho napäťia a prípravu pre pripojenie programátora. Rozloženie pinov týchto dosiek je takmer zhodné a je ho možné vidieť na obrázku 3.1.



Obr. 3.1: Rozloženie pinov kitu Blackpill [17]

4 Návrh Hardware

Počas semestrálneho projektu bola navrhnutá prvá verzia zapojenia a na základe nej bol vyrobený plošný spoj. Ten bol osadený a robili sa na ňom prvé merania a experimenty. Keďže tento návrh obsahoval nedostatky bola navrhnutá druhá verzia. V druhej verzii bol odstránený USB/RS232 prevodník realizovaný pomocou obvodu FTDI FT232RL. Tento obvod mal slúžiť pre komunikáciu s PC. Keďže mikrokontrolér má implementované USB a použitá doska obsahuje USB konektor bol pre komunikáciu použitý ten. V prvej verzii boli pripravené viaceré filtre v obvode spracovania nízkofrekvenčného signálu. Počas testov a ladenia tieto filtre neboli potrebné a preto boli odstránené. Okrem zosilňovača analógového signálu bola do obvodov spracovania nízkofrekvenčného signálu pridaná časť, ktorá prevádza signál na obdĺžnik a privádza ho na ďalší zo vstupov mikrokontroléra, kde sa nemeria pomocou analógovo digitálneho prevodníka úroveň a tvar signálu, ale len frekvencia signálu.

4.1 Napájanie

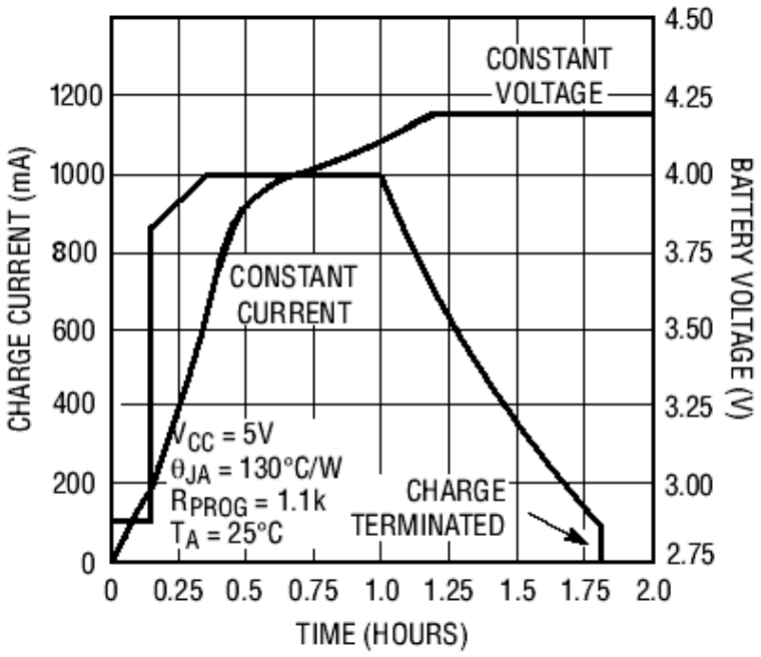
Keďže zariadenie malo byť schopné pracovať nezávisle potrebuje zdroj energie. Ako zdroj energie bol zvolený Li-ion článok formátu 18650 [18]. Ide o populárny v praxi veľmi často používaný článok v notebookoch, elektrické kolobežky. Li-ion články majú výborný pomer rozmer/kapacita hmotnosť kapacita. Ako vstup pre nabíjačku článku je zvolený 5V z USB portu. Je využitý USB-C port obsiahnutý v riadiacej doske Blackpill. USB port je štandardne používaný v nabíjačkách pre komerčne dostupné zariadenia ako napríklad telefóny, tablety alebo poweebanky, a ponúka teda širokú kompatibilitu. Pre riadenie nabíjacieho procesu je použitý obvod TP4056 [18]. Zariadenie je navrhnuté tak, že sa dá použiť ľubovoľný článok Li-Ion, LiPo s koncovým nabíjacím napätím 4,2 V. K zariadeniu nie je možné pripojiť článok LiFePo pretože ten má koncové nabíjacie napätie do 3,6 V. Veľkosť nabíjacieho prúdu v prúdovom režime je možné definovať pomocou rezistora R_{PROG} podľa vzťahu [18]

$$I_{BAT} = \frac{1\ V}{R_{PROG}} 1200. \tag{4.1}$$

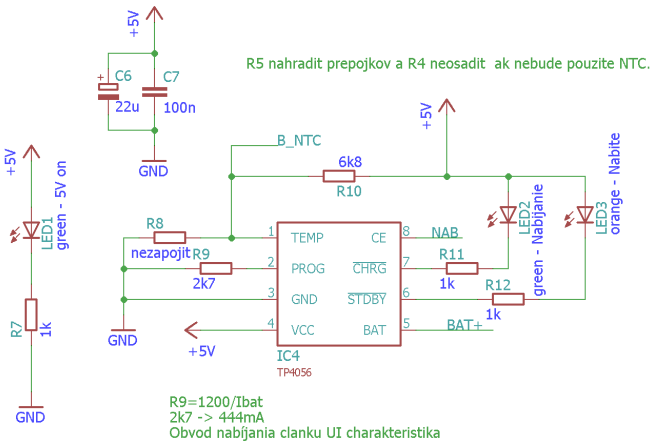
Pre tento projekt bol zvolený rezistor R9 s hodnotou 2,7 kΩ, čo zodpovedá nabíjaciemu prúdu I_{NAB} 444mA. Veľkosť tohoto prúdu je odporúčaná výrobcom ako 0,2 násobok kapacity akumulátora a bola určená ako [18]

$$I_{NAB} = 0,2 * 2200 = 440\ mA. \tag{4.2}$$

Obvod pritom využíva štandardnú voltámpérovú charakteristiku zobrazenú na obrázku 4.1.



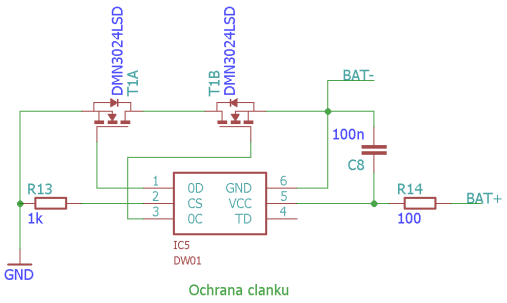
Obr. 4.1: V-A charakteristika cyklu nabíjania Li-ion článku[19]



Obr. 4.2: Schéma zapojenia obvodu nabíjania

Pokiaľ je článok vybitý má tendenciu odoberať z nabíjača vysoký prúd úmerný vnútornému odporu článku a vnútornému napätiu. Tento prúd je obvodom obmedzovaný na nastavenú hodnotu. Tento mód práce nabíjača sa nazýva režim konštantného prúdu. Akonáhle sa napätie na článku priblíži hodnote 4,2 V už nesmie elektronika dovoliť ďalší nárast napätia na článku. Aby nedošlo k prekročeniu nabíjacieho napätia obvod začne pracovať v móde konštantného napätia a teda nedovolí prekročiť nabíjacie napätie (správa sa ako zdroj napätia). Články Li-Ion častokrát umožňujú nabíjanie a vybíjanie veľkým prúdom čo môže spôsobiť ich výrazné zohriatie. Preto sú často doplnené o snímanie teploty článkov, aby nedošlo k ich zničeniu alebo k

zničeniu zariadenia. Dôležité je meranie teploty v prípade poškodeného článku. Poškodené články pri nabíjaní generujú značné teplo na čo musia nabíjacie obvody reagovať prerušením nabíjania. Použitý riadiaci obvod obsahuje spätnú väzbu pre NTC. Pomocou tohto vstupu je nastavený teplotný rozsah nabíjania článku (spodná aj horná teplota). V pracovnom rozsahu teplôt musí byť napätie na vstupe TEMP v rozsahu 45% - 80% napájacieho napätia. V prípade ak nechceme použiť spätnú väzbu je zapojenie doplnené o prepojku pomocou ktorej je vstup TEMP spojený s GND čo spôsobuje deaktiváciu tejto ochrannej funkcie. Mód v akom obvod pracuje respektíve v akom stave je článok signalizujú dve LED diódy. Dióda LED2 indikuje že prebieha nabíjanie a dióda LED3 indikuje že článok je už nabitý. Spínačom S5 (zopnutý kontakt 1 a 2) sa povoľuje pomocou vstupu CE chod nabíjača. Keď je na tento vstup privedené napätie +5 V nabíjač pracuje ak je privedených 0 V cez 1 kΩ rezistor nabíjač je blokovaný. O ochranu článku sa stará aj obvod DW01-P [20]. Ide o ochranný obvod obsahujúci detekciu prebitia, podbitia, skratu, preťaženia článku a detekciu nabíjača. Tento obvod je v prípade detekcie nevhodného stavu schopný odpojiť článok pomocou dvoch MOSFET tranzistorov T1 a T2.



Obr. 4.3: Schéma zapojenia ochranného obvodu batérie

Napätie na Li-Ion článku sa podľa stavu nabitia môže pohybovať v rozsahu 3,0 – 4,2 V [18]. Pre napájanie obvodov je potrebné stabilné napätie 5 V. Pre zabezpečenie téjto podmienky je v zariadení zvyšujúci menič, ktorý z napätia článku vytvorí stabilných 5 V. Pre jeho zapnutie je nutné dať spínač do polohy kde má prepojené kontakty 2 a 3. Zvyšujúci menič je riadený obvodom MT3608 [21]. Tento obvod dokáže pracovať s napätiami od 2,0 V. Pracuje s pevnou spínacou frekvenciou 1,2 MHz. Výstupné napätie obvodu je nastaviteľné pomocou deliča tvoreného rezistormi R15

a R16. Veľkosť týchto rezistorov bola určená výpočtom

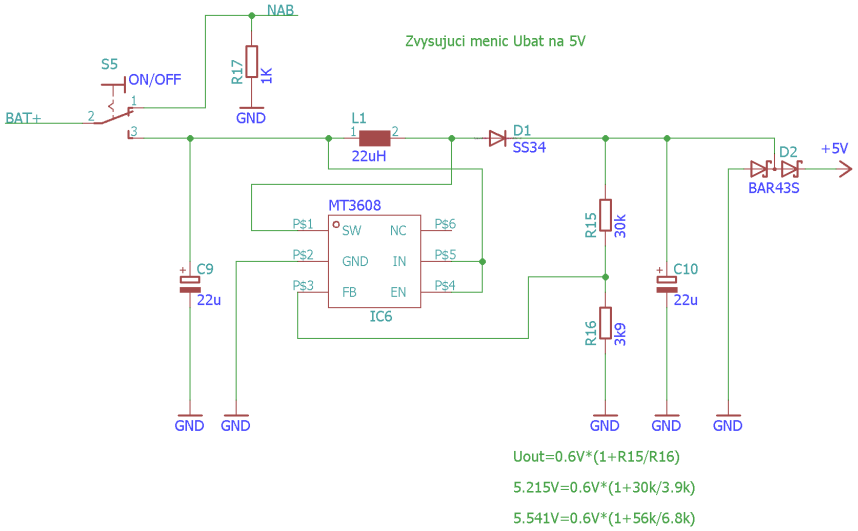
$$U_{OUT} = U_{REF}\left(1 + \frac{R_{15}}{R_{16}}\right) \tag{4.3}$$

$$\frac{U_{OUT}}{U_{REF}} - 1 = \frac{R_{15}}{R_{16}} \tag{4.4}$$

$$\frac{5}{0,6} - 1 = \frac{R_{15}}{R_{16}} \tag{4.5}$$

$$\frac{22}{3} = \frac{R_{15}}{R_{16}} = \frac{110k\Omega}{15k\Omega}. \tag{4.6}$$

Obvod pre svoju činnosť nepotrebuje externý polovodičový spínací prvok, ten je obsiahnutý priamo v integrovanom obvode (výkonový MOSFET). Prúdový limit tohoto integrovaného spínača je 4 A. Súčasťou obvodu je aj tepelná ochrana proti preťaženiu. Prúd spínačom je meraný priamo v obvode a použitý v regulácií v podriadenej prúdovej slučke. Druhý zosilňovač tvorí nadriadenú napäťovú slučku regulácie výstupu meniča. Pre korektnú funkciu meniča je nutné použiť indukčnosť. Hodnotu indukčnosti výrobca obvodu odporúča v rozsahu 4,7 až 22 µH s malým jednosmerným odporom (DCR). Keďže obvod pracuje na vysokej spínacej frekvencií tak je požiadavkou na cievku aby mala mále straty v jadre. Vzhľadom na vysokú spínaciu frekvenciu musí byť použitá Schottkyho dióda. Použitá je dióda SS32, ktorá je určená pre aplikácie DC/DC meničov. Pracovné napätie má do 20 V a prúd až 3 A. Vstup aj výstup je filtrovaný pomocou kondenzátorov C9, C10 s kapacitou 22 µF podľa odporúčania výrobcu.



Obr. 4.4: Schéma zapojenia zvyšujúceho meniča na 5V

Na vstup PB0 je privedené cez delič R5 a R6 prispôsobené napätie článku pre potreby monitorovania stavu Li-Ion článku. Keďže napätie na článku môže dosiahnuť

až 4,7 V je toto napätie upravené aby neprekročilo napätie, ktoré je v merateľnom rozsahu ADC Mierka pre prevodník je:

$$U = \frac{R_6}{R_5 + R_6} U_{BAT} \tag{4.7}$$

$$U = \frac{3300}{2200 + 3300} U_{BAT} \tag{4.8}$$

$$U = \frac{33}{55} U_{BAT} \tag{4.9}$$

Paralelne s rezistorom R6 je pripojený filtračný kondenzátor C5 slúžiaci na potlačenie zvlnenia a iných rušení. LED 1 je pripojená na napäťovú hladinu 5 V a signalizuje prítomnosť napätia na tejto hladine. Kondenzátory C6, C7 filtrujú napájacie napätia pre obvod nabíjania IC4 a C11, C12 pre napájanie operačného zosilňovača IC2.

4.2 Spracovanie NF signálu

Zariadenie ma za úlohu spracovať nízkofrekvenčný signál obsahujúci dáta. Vstupný nízkofrekvenčný signál je nutné prispôsobiť pre potreby merania pomocou ADC vstupu alebo digitálnych vstupov. ADC vstup môže snímať napätie v rozsahu 0 až 3 V a podobne aj digitálny vstup. Nízkofrekvenčný audio signál je možné do zariadenia pripojiť pomocou štandardného 3,5 mm audio jacka (X1) alebo pomocou cinch konektora (X2). Voľba typu konektora je možná pomocou prepojky JP1. U konektoru X1 je možné voliť pomocou prepojky JP2 či bude spracovaný pravý alebo ľavý kanál privedeného nf signálu.

Úroveň vstupného analógového nízkofrekvenčného signálu je nastaviteľná pomocou napäťového deliča tvoreného potenciometrom R32. Takto upravený signál sa následne rozdeľuje do dvoch ciest. V prvej ceste signál prechádza cez väzobný kondenzátor C11. Pomocou dvojitej Schottkyho diódy D4 je vstupná elektronika chránená proti prepätiam keďže vstup môže byť odpájaný a pripájaný obsluhou, čo môže byť zdrojom rôznych prepätí. Signál je zosilnený investujúcim zosilňovačom a privedený na vstup mikrokontroléra na ďalšie spracovanie pomocou analógovo digitálneho prevodníka. Pomocou rezistorov R19 a R20 možno zvoliť zosilnenie obvodu. Pre zosilnenie platí:

$$K(p) = -\frac{R_{19}}{R_{20}}, \tag{4.10}$$

$$K(p) = -\frac{100k}{33k}, \tag{4.11}$$

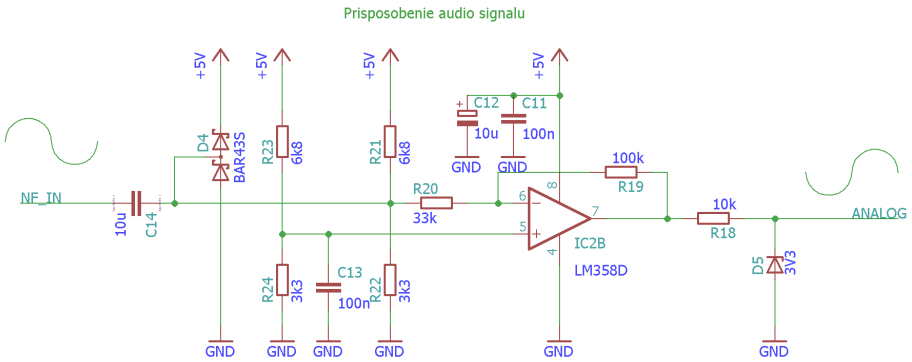
$$K(p) = -3,03. \tag{4.12}$$

Na neinvertujúci aj invertujúci vstup je privedené pomocou napäťových deličov napätie 1,635 V čo je nutné vzhľadom na to, že operačný zosilňovač nie je napájaný zo symetrického zdroja a zároveň zabezpečuje posunutie signálu do stredu pracovného rozsahu ADC vstupu. Kondenzátor C13 potláča zákmity na neinvertujúcom vstupe.

$$U = 5V\frac{R_24}{R_{23} + R_{24}}, \tag{4.13}$$

$$U = 5V\frac{3k3}{6k8 + 3k3}, \tag{4.14}$$

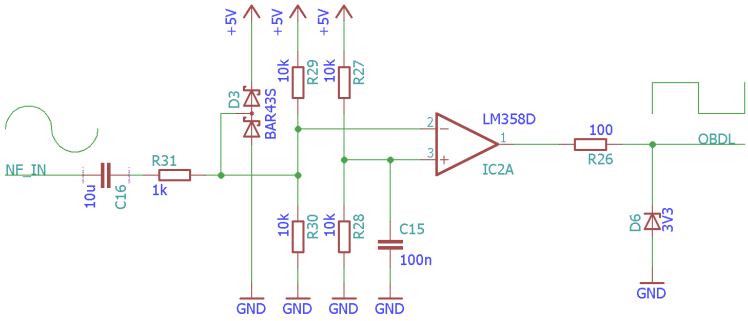
$$U = 1,635V. \tag{4.15}$$



Obr. 4.5: Schéma zapojenia zosliňovača na vstupe PA1

Takto upravený a zosilnený signál je cez rezistor R18 privedený na vstup PA1 mikrokontroléra. Rezistory R18 a R17 môžu tvoriť napäťový delič a znížiť signál tak, aby neprekročil napätie 3V pri plnom vybudení operačného zosilňovača. Druhým elegantnejším riešením je zapojenie zenerovej diódy D3 so zenerovým napätím 3,3 V miesto rezistora R17. Riešenie pomocou zenerovej diódy je nezávislé od napájacieho napätia operačného zosilňovača. Takto spracovaný signál je následne vzorkovaný pomocou ADC prevodníka v mikrokontroléri. Signál na vstupe mikrokontroléra je invertovaný (posunutý o 180°), to ale na vyhodnocovanie nemá žiadny vplyv. V druhej ceste je zase signál zapojením operačného zosilňovača ako komparátor prevedený na obdĺžnikový signál uľahčujúci použitie čítača a časovača na určovanie frekvencie signálu. Na neinvertujúci aj invertujúci vstup je privedené pomocou napäťových deličov napätie 1,635 V čo je nutné vzhľadom na to, že operačný zosilňovač nie je napájaný zo symetrického zdroja. Na invertujúci vstup je signál privedený cez väzobný kondenzátor C16. Prechádza cez obmedzovací rezistor R31. Pomocou dvojitej Schottkyho diódy D3 je vstupná elektronika chránená proti prepätiam keďže vstup môže byť odpájaný a pripájaný obsluhou, čo môže byť zdrojom rôznych prepätí. Takto upravený signál je cez rezistor R26 privedený na vstup PA0

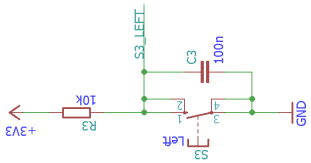
mikrokontroléra. Rezistory R26 a R25 môžu tvoriť napäťový delič a znížiť signál tak aby neprekročil napätie 3V pri plnom vybudení operačného zosilňovača. Taktiež ako v predchádzajúcom je elegantnejším riešením zapojenie zenerovej diódy D6 so zenerovým napätím 3,3 V miesto rezistora R25. Takto spracovaný už obdĺžnikový signál je následne spracovaný mikrokontrolérom. Hodnotu rezistora R26 je potrebné zvoliť malú napríklad 100 Ω, pretože pri veľkom odpore bude hrana obdĺžnikového signálu pomalá (vplyvom kapacít vznikne lichobežník).



Obr. 4.6: Schéma zapojenia komparátoru pre vstup PA0

4.3 Klávesnica

Interakcia užívateľa so zariadením je možná pomocou štyroch tlačidiel (S1-S4). Pomocou týchto tlačidiel je možné sa pohybovať v ponuke menu a nastavovať základné parametre. Keď je tlačidlo stlačené na vstupe mikrokontroléra je 0 V. K tlačidlám sú pripojené paralelne 100nF keramické kondenzátory, ktoré potláčajú zákmity a rušenia vznikajúce pri stlačení a pustení tlačidla. K tlačidlu je pripojený rezistor, ktorý definuje úroveň signálu (3,3 V) na vstupe mikrokontroléra, keď nie je stlačené tlačidlo (pull-up rezistor). Tlačidlá sú ošetrené voči zákmitom aj softvérovo.



Obr. 4.7: Schéma zapojenia tlačidla

4.4 LCD displej a I2C

Na základnú vizualizáciu nastavení a prijatých dát sa používa štvorriadkový alfanumerický LCD display s 20 znakmi na riadok. Display je pripojený na I2C zbernicu a teda potrebuje len dva dátové signály (SDA, SCL). Štandardný alfanumerický display je možné ovládať pomocou troch riadiacich signálov (Register Select, Read/Write, Enable) a 8 dátových. V praxi sa ale častejšie používa na počet signálov menej náročne modifikované zapojenie so štyrmi dátovými signálmi (štvorbitové zapojenie). V mojej aplikácií riadiace aj dátové signály realizuje obvod PCF8574. Tento obvod je expander portov pre I2C zbernicu. Obsahuje 8 pinov pričom každý z pinov môže byť vstupom aj výstupom. Nastavenie môže vykonať master obvod na I2C zbernici prestavením registra obvodu. Mikrokontroler predstavuje mastra na I2C linke a obvod PCB8574 slave. Obvod disponuje tromi digitálnymi vstupmi (A0, A1, A2) pomocou ktorých je definovaná adresa obvodu na I2C zbernici. Nastavenie vstupov A0, A1, A2 je nutné brať v úvahu pri obsluhe displeja v programe.

4.5 Plošný spoj

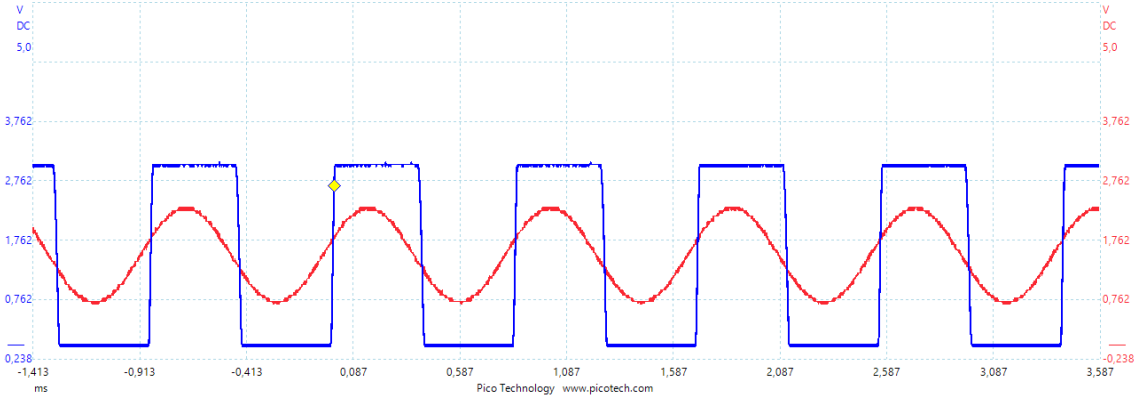
Na základe návrhu popísaného v tejto kapitole bola zostavená v programovom prostredí Eagle schéma obvodu. Po jej dokončení bol v rovnakom programovom prostredí vytvorený návrh dosky plošného spoja. Bola navrhnutá v podobe dvojvrstvového plošného spoja s prekovenými dierami. Pri návrhu bola snaha voliť súčiastky prevažne vo forme SMD. Pre rezistory a kondenzátory bolo volené prevedenie púzdra 1206. Keďže súčiastky budú osádzané ručne a s obmedzeným vybavením, bola snaha vyhnúť sa súčiastkám v príliš malom prevedení. Pomocou programu Eagle boli nakoniec vygenerované gerber súbory potrebné pre výrobu. Doska plošných spojov bola na základe vygenerovanej dokumentácie profesionálne vyrobená.

4.6 Sprevázdkovanie plošného spoja

Na doske plošného spoja boli najprv osadené mechanické prvky ako lišty, konektory, prepínač a tlačidlá, ktoré nie sú citlivé ani na teplo ani na nešetrnú manipuláciu. Následne boli osadené prvky zvyšujúceho meniča. Po osadení bolo privedené na vstup napätie 3,3 V. Na výstupe zdroja bolo kontrolované výstupné napätie. Toto napätie dosahovalo 5,2 V čo bola požadovaná hodnota.

Následne bol zapojený obvod nabíjania. Prepnutý prepínač do druhej polohy. Na vstup obvodu bolo privedené napätie 5V a na výstupe bolo kontrolované nabíjacie napätie. Toto napätie neprekročilo 4,2 V. Napäťový mód pracoval správne. Následne bola pripojená batéria. Bol skontrolovaný nabíjací prúd nastavený rezistorom R9.

Prúd bol 444 mA čo odpovedá navrhnutej hodnote. Prúdový mód pracuje správne. Taktiež bola kontrolovaná korektná signalizácia jednotlivých módov led diódami (LED 2 nabíjanie, LED 3 nabitý článok). Ďalej boli osadené zvyšné súčiastky, skontrolované napäťové úrovne jednotlivých deličov napätia. Po finálnej kontrole napätia v pätici pre mikrokontrolér, do nej bol pripojený modul s mikrokontroleérom. Nakoniec bol na vstup zariadenia privedený nízkofrekvenčný signál a pomocou osciloskopu boli skontrolované priebehy na pinoch PA0 a PA1.



Obr. 4.8: Priebehy získané z pinu PA0(modrý) a PA1(červený).

5 Komunikačný protokol

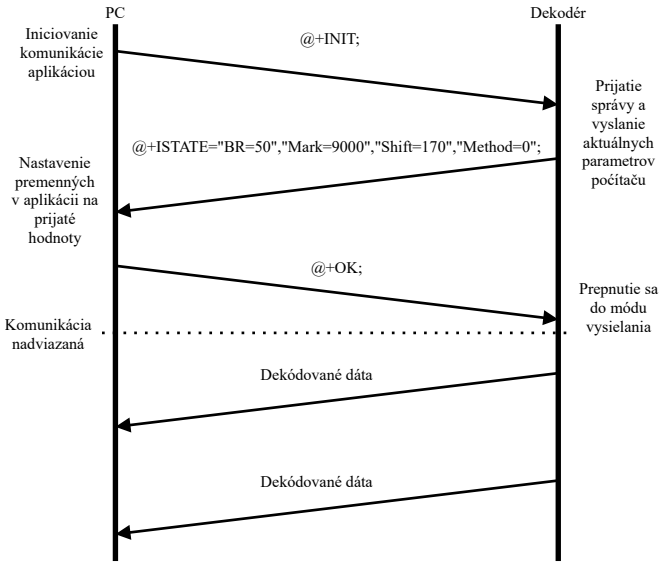
Zariadenie a počítačová aplikácia, ktoré sú súčasťou tejto práce by mali vzájomne komunikovať. Komunikácia by pritom mala primárne zabezpečovať odosielanie dekódovaných dát zo zariadenie do PC aplikácie a nastavovanie jednotlivých parametrov zariadenia pomocou pc aplikácie. Aby bola štruktúra komunikácie jednotná bola vytvorená séria príkazov, ktoré by sa dali označiť za jednoduchý protokol. Forma týchto príkazov je pritom inšpirovaná AT príkazmy z Hayesovej prikazovej sady, ktorá sa používa napríklad pri GSM moduloch, modemoch a GPS moduloch. Všetky správy posielané medzi zariadením a aplikáciou začínajú symbolom @. Symbol @ bol zvolený z dôvodu že sa nenachádza v abecede, ktorú používa RTTY a je ho teda v prijatých dátach možné jednoduchšie detekovať. Koniec každého príkazu je indikovaný symbolom ;.

Na realizáciu komunikácie bol použitý USB-C port ktorý sa nachádza na doske Blackpill. Predpokladá sa že detekcia chýb a prípadné znovuzaslanie dát bude zabezpečené na úrovni USB komunikácie na základe CRC a potvrdzovania v poobe ACK príznakov. Komunikácia zároveň bude prebiehať len na krátku vzdialenosť cez pripojený USB kábel. Chybovosť by tým pádom mala byť nízka. Vytvorený komunikačný model sa teda nebude zaoberať detekciou a opravou chýb prijatých dát. Využíva sa v ňom ale potvrdzovanie prijatia správy, najmä za účelom overovania pripojenia zariadenia a detekcie prípadného vzniku chyby, o ktorej môže následne byť užívateľ upozornený prostredníctvom aplikácie.

5.1 Priebeh komunikácie

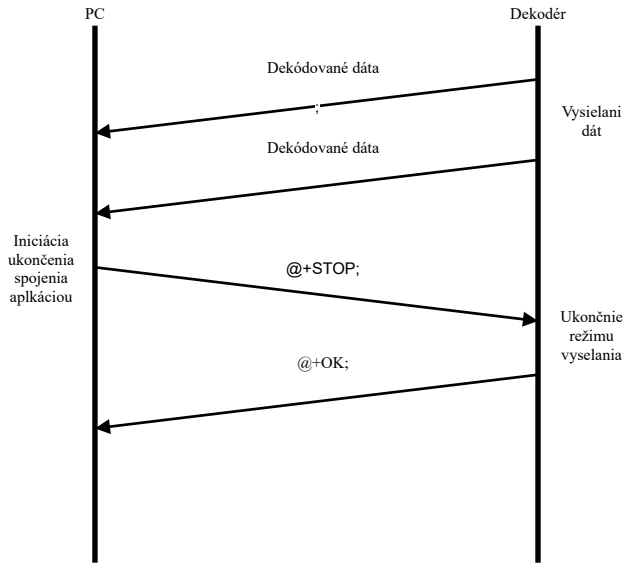
Komunikácia medzi zariadením a aplikáciou je vždy iniciovaná zo strany aplikácie. Užívateľ v aplikácii určí port na ktorom by komunikácia mala prebiehať. Na tento port sa vyšle príkaz @+INIT. Ak je na tomto porte pripojené naše zariadenie, zareaguje vyslaním príkazu @+ISTATE, ktorý bude obsahovať nastavené parametre zo zariadenia. Zariadenie sa po prijatí príkazu INIT tiež prepne do módu v ktorom sa deaktivujú tlačidlá a začne na USB port vysielať dekódované dáta. Tie sú posielané mimo štrukturovaných správ, iba ako čisté dáta. Aplikácia teda od nadviazania vyhodnocuje všetky prijaté znaky mimo príkazov ako dekódované dáta, ktoré môže vypisovať.

Komunikácia môže byť zo strany aplikácie ukončená pomocou príkazu @+STOP. Tento príkaz nastaví zariadenie späť do bežnej prevádzky, odblokuje jeho ovládanie a prestane posielať dekódované dáta. Môže dôjsť k situácii kedy je zariadenie od počítača odpojené pred poslaním packetu STOP. Toto odpojenie je možné zo strany počítača detekovať a reagovať naň. Naše zariadenie túto možnosť však nemá a mohlo



Obr. 5.1: Diagram nadviazania spojenia

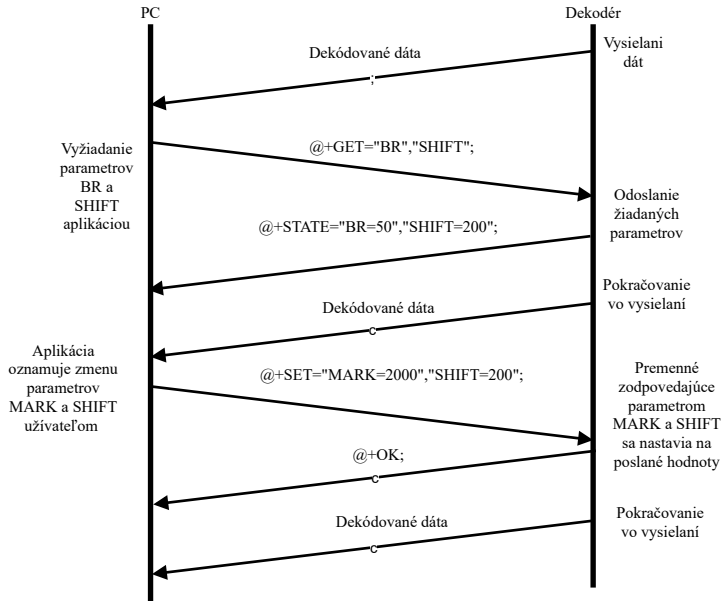
by dôjsť k tomu, že zostane zaseknuté v režime odosielania dát s deaktivovanými ovládacími prvkami. Túto situáciu by bolo potrebné riešiť reštartom zariadenia. Aby sa jej predišlo, sleduje zariadenie intervaly medzi prijatými správami od aplikácie. V prípade že tento interval presiahne preddefinovanú hodnotu, ktorá je v tomto prípade nastavená na 0,5 s, zariadenie pošle správu v tvare @+TRY. Ak je naďalej pripojené k počítaču s bežiacou aplikáciou, dostane odpoveď v tvare @+OK. Znovu čaká a ak nedostane odpoveď proces zopakuje. Po vyslaní 6 správ bez odpovede sa od zariadenie odpojí a prestane vysielať dáta.



Obr. 5.2: Diagram ukončenia komunikácie

Užívateľ môže následne pomocou aplikácie prestavovať jednotlivé parametre zariadenia. Na nastavenie parametrov slúži príkaz @+SET, ktorý je nasledovaný identifikátorom, ktorý indikuje aký parameter sa má nastaviť a novou hodnotou tohoto parametra. V rámci tohoto príkazu je možné zmeniť aj niekoľko parametrov súčasne. Zariadenie zareguje na prijatie tohoto príkazu potvrdením v tvare @+OK.

Aplikácia si taktiež dokáže vyžiadať práve nastavené dáta pomocou príkazu @+GET, nasledovaný identifikátormi, ktoré určujú žiadané parametre. Odpoveďou je príkaz @+STATE nasledovaný danými parametrami podobne ako príkaz ISTATE.



Obr. 5.3: Diagram získania a nastavenia parametrov zariadenia

Zariadenie taktiež ponúka užívateľovi možnosť jednu frekvenciu zapísať do pamäte flash. Príkaz, ktorý indikuje aká frekvencia má byť zapísaná má tvar @+WRITE, ktorý je nasledovaný údajom o hodnote, ktorá sa má zapísať.

6 Program pre Mikrokontrolér

Program pre mikrokontrolér je možné realizovať viacerými spôsobmi. Je ho možne písať priamo v inštrukčnej sade mikrokontroléra. Program realizovať v jazyku C/C++ použitím LL funkcií. Program realizovať v jazyku C/C++ s použitím HAL funkcií. Kontroléry STM majú podporu aj pre operačné systémy tak je možné program realizovať v operačnom systéme napríklad Mbed OS. Použiť prostredie Arduino. Každé z riešení ma svoje výhody a nevýhody. Z hľadiska komfortu pre programátora, postačujúcej rýchlosti a efektivite výsledného programu bolo zvolená platforma Arduino.

6.1 Platforma Arduino

Pre tvorbu softvéru mikrokontroléra bolo zvolené vývojové postredie Arduino IDE a k nemu patriace knižnice. Arduino predstavuje open-soure platformu, ktorá umožňuje jednoduchý vývoj embedded systémov, vďaka čomu má širokú komunitu a je na ňu dostupné veľké množstvo knižníc. Táto platforma sa obvykle používa spolu doskami Arduino, ktoré sú založené na rôznych radách mikrokontrolérov od spoločnosti ATmega. Aby bolo možné toto prostredie používať v kombinácii s mikrokontrolérmi STM32, bolo potrebné doinštalovať knižnice . Program je písaný v jazyku C++, pričom je možné využívať časti kódu založené na HAL funkciách STM mikrokontrolérov. Program pozostáva z dvoch hlavných funkcií Setup a Loop. Funkcia setup sa vykonáva raz, vždy po zapnutí alebo reštartovaní mikrokontroléru. Mala by obsahovať primárne úvodné nastavenie hardvéru a inicializáciu premenných. Funkcia loop predstavuje nekonečnú slučku, ktorá sa cyklicky vykonáva po celú dobu, čo je mikrokontrolér zapnutý. Kód v tele tejto funkcie určuje čo by mal mikrokontrolér vykonávať a predstvuje teda obdobu funkcie main [22]. Ďalšou časťou programu sú definície obsluhy prerušení a vlastné funkcie. Z dôvodu prehľadnosti a logického oddelenia bol kód štrukturovaný do viacerych súborov, kde každý zodpovedá jednej knižnici.

Kompilácia je zabezpečená postredím Arduino. Toto prostredie taktiež podporuje zavádzanie programu do mikrokontroléra. Alternatívne umožňuje vytvoenie binárneho súboru, ktorý následne môže byť do mikrokontroléru zavedený pomocou softwéru určeného pre STM32. Program je do kontroléru zavádzaný pomocou USB programátora ST-Link V2.

6.2 Demodulácia signálu

Demodulácia BFSK signálu je založená na detekcii dvoch frekvencií zodpovdajúcich log. 1 a log 0. Teoretický popis BFSK klúčovania, jeho demodulácie a niektoré

metódy sú bližšie rozobrané v kapitole 2.2.

Zhotovenie plošného spoja umožňuje ako vstup plošného spoja použiť pin PA0, na ktorý je privedený priamo analógový signál alebo vstup PA1, na ktorý je pripojnený obdĺžnikový signál predspracovaný operačným zosilňovačom. K získaniu hodnoty frekvencie z týchto signálov je potrebné pristupovať zásadne odlišnými spôsobmi. Program obsahuje niekoľko metód na demoduláciu signálu. Pred demoduláciou je vhodné si určiť niekoľko parametrov signálu. Jeho periódu

$$T = \frac{1}{f}, \tag{6.1}$$

dĺžku trvania jedného symbolu T_0, ktorú vypočítame ako

$$T_0 = \frac{1}{M}, \tag{6.2}$$

kde M je symbolová rýchlost v Baudoch. Keďže pri prenose dát môže dochádzať k vzniku chýb, či už z dôvodu rušenia v signále alebo časovania mikrokontroléru, je vhodné pre každý signálový prvok detekovať jeho frekvenciu viacnásobne. Perióda merania frekvencie teda bude zodpovedať

$$T_m = \frac{1}{Mn}, \tag{6.3}$$

kde n je počet meraní za periódu jedného symbolu.

6.2.1 Funkcia pulseIn

Prvá, najjednoduchšia metóda demodulácie využíva pre svoju funkciu obdĺžnikový signál. Perióda tohoto signálu by mala zodpovedať perióde analógového signálu na vstupe zariadenia. Tvar tohoto signálu však umožňuje jednoduchšie zmeranie periódy tohoto signálu dostupnými prostriedkami kontroléru.

Jedným zo spôsobov zmerania periódy signálu je funkcia pulseIn. Tejto funkcii je možné pomocou jej argumentov špecifikovať pin a logickú hodnotu, na ktorú čakáme. Tretí, voliteľný parameter je timeout, alebo maximálny čas v μs, ktorý funkcia čaká na začiatok impulzu. Predvolená hodnota tohoto času je 2 s. Po zavolaní funkcie, zariadenie čaká kým pin prejde do požadovanej logickej hodnoty a začne časovať, až do ďalšej zmeny logickej hodnoty na pine. Po dokončení časovania táto funkcia vráti dobu trvania impulzu v μs. Dokáže pritom zmerať pulzy s dĺžkou od 10 μs do 3 minút, pričom ale presnosť pri vyšších dĺžkach klesá [23].

Zavolaním tejto funkcie v sekvencii, dvakrát po sebe, vždy pre inú logickú hodnotu, dokážeme určiť periódu signálu v danom momente. Z hodnoty periódy je následne pomocou vzorca možné vypočítať frekvenciu signálu.

V prvej verzii sa táto funkcia volala periodicky v hlavnej slučke programu. Dochádzalo však ku konfliktu s niektorímy dlhšie trvajúcimi funkciami, najmä s vykreslovaním na displej a časová perióda medzi volaniami funkcie pulseIn nebola konštaná. Z tohoto dôvodu dochádzalo k častým chybám pri vyhodnocovaní prijatých bitov. Funkcia bola následne premiestnená do prerušenia. Vykonávanie funkcie v rámci prerušenia eliminovalo konflikt s dlhšie sa vykonávajúcimi funkciami, keďže sa mohla vykonať aj počas ich priebehu. Hlavnou nevýhodou použitia tejto funkcie je, že mikrokontrolér počas čitania hodnoty z pinu nemôže vykonávať iné časti programu. Dochádza teda k určitému plýtvaniu výpočetnými prostriedkami mikrokontroléru [23].

6.2.2 Čítanie pulzov pomocou prerušenia (ISR)

Táto metóda demodulácie bola navrhnutá ako efektívenjšia alternatíva metódy pulseIn. Použitý mikrokontrolér ponúka možnosť priradiť k digitálnemu vstupu prerušenie. Toto prerušenie sa vykoná vždy keď na zvolenom pine dôjde k určitej udalosti. Prerušenie sa pritom môže vykonávať keď sa napine vyskytne logická hodnota 0, zmena logickej hladiny, stúpajúca hrana alebo klesajúca hrana signálu. Pri nastavení na stúpajúcu hranu by sa toto prerušenie malo vykonávať s rovnakou periódou ako je perióda obdĺžnikového signálu na vstupe.

Pri každom vyvolaní prerušenia si do premennej uložíme aktuálny čas. V nasledujúcom vyvolaní tento tento čas odpočítame od aktuálneho času a uložený čas prepíšeme aktuálnym. Hodnota získaná týmto výpočtom by mala prestavovať periódu signálu. Pre získanie frekvencie signálu stačí, podobne ako pri pulseIn, už len spraviť prevrátenu hodnotu z tohoto čísla. Narozdiel od použitia pulseIn nieje procesor pri tejto metóde počas merania periódy zaneprázdnený a dokáže zatiaľ vykonávať iné funkcie.

6.2.3 FFT

Na demoduláciu BFSK je možné tiež použiť algoritmus Rýchlej Fourierove transformácie, ktorá je bližšie popísaná v časti 2.2.4. V tomto prípade bola konkrétne použitá knižnica arduinoFFT.h, ktorú je možné získať na [24].Spracovanie signálu pomocou FFT sa skladá z niekoľkých krokov. Najskôr je potrebné z analógového vstupu získať potrebný počet vzorkov N, na základe ktorých budeme FFT počítať. Hodnota N by v ideálnom prípade mala byť mocninou čísla 2. Výpočet FFT je relatívne náročný. Dĺžka jej výpočtu rastie spolu s počtom vzorkov, z ktorých je počítaná. Po testovaní rôznych veľkostí N bola ako najvhodnejšia zvolená hodnota 64 s vzorkovacou frekvenciou 16384 Hz. Pri týchto parametroch trvá výpočet FFT 8 ms. Keďže v programe meriame hodnotu každého symbolu v signále dvakrát, je možné

touto metódou demodulovať signál s trvaním jedného symbolu maximálne 16 ms, čo zodpovedá symbolovej rýchlosti 62,5 Bd. Štandardom pre RTTY sú rýchlosti 45 a 50 Bd, takže táto rýchlosť je považovaná za dostatočnú. Pri vyšších rýchlostiach dĺžka výpočtu presahuje dĺžku periódy signálového prvku, a bolo by teda potrebné buď znížiť počet vzorkov alebo každý symbol merať len raz.

Ďalším krokom je na získané vzorky aplikovať algoritmus FFT. Je pritom použitá funkcia compute zahrnutá v knižnici arduinoFFT.h. Výsledkom tohoto výpočtu sú dve polia, jedno reprezentujúce reálnu a druhé imaginárnu časť výsledku. Z týchto dát je následne pomocou funkcie ComplexToMagnitude možné z týchto dát získať magnitúdy frekvencií, na základe ktorých, je možné určiť ich výskyt v signále. Frekvencia s najvyššou magnitúdou je z týchto dát získaná pomocou funkcie MajorPeak.

Porovnaním tejto frekvencie s hľadanými frekvenciami s určitou toleranciou, je možné určiť, či daný signálový prvok zodpovedá logickej 1 alebo 0.

6.3 Dekódovanie signálu

Po demodulácii je každý symbol originálneho signálu reprezentovaný sekvenciou n po sebe nasledujúcimi bitmi, ktoré je potrebné vhodne zjednodušiť do formy, v ktorej každému vstupnému symbolu zodpovedá jeden bit v programe. To je založené na použití dvoch premenných, ktoré zaznamenávajú počet prijatých logických jednotiek a núl v sekvencii. Po prijatí n bitov sa tieto premenné porovnajú a hodnota výsledného bitu bude zodpovedať logickej hodnote, ktorá bola napočítaná viackrát. V prípade ak je n rovné párnemu číslu môže dôjsť k situácii že sa tieto premenné budú rovnať. Obvykle k nej dochádza na rozhraní dvoch po sebe nasledujúcich symbolov. V tomto prípade pred vyhodnotením zariadenie počká na ďalší bit, ktorý jednoznačne rozhodne o hodnote. Premenné zaznamenávajuce počty logických hodnôt sa po každom vyhodnotení vynulujú.

Výsledkom by mali byť bity prijímaného RTTY signálu, ktoré je potrebné dekódovať. RTTY dáta sa skladajú z troch častí

- Start bit – logická 0,
- Dátové bity – 5 bitov reprezentujúcich znak v Baudotovom kóde,
- Stop bit – 1 1,5 alebo 2 bity s logickou hodnotou 1,

Z prijatých dát je teda potrebné najskôr extrahovať dátové bity nesúce informáciu o znaku. Program najskôr čaká na prijatie logickej nuly, ktorá reprezentuje Start bit a teda začiatok RTTY dát. Logické 1 prijaté pred nájdením start bitu niesú ďalej spracované a sú zahodené. Po prijatí logickej 0, uloží program 5 nasledujúcich bitov do poľa. Predtým, než tieto dáta vyhodnotí overí si či po nich nasledujúce bity zodpovedajú danému počtu stop bitov s logickou hodnotu 1. V prípade, že tieto bity stop bitom nezodpovedajú, program najskôr prehľadá uložené prijaté dáta a pokúsi

sa v nich nájsť ďalšiu nulu. Ak je v dátach nula nájdená, je považovaná za nový start bit a proces sa opakuje. Po nájdení zodpovedajúcich start a stop bitov môžeme bity uložené v poli považovať za dátové bity Baudotovho kódu.

Aby bolo možné s týmito bitmi jednoduchšie pracovať sú v programe reprezentované dátovým typom byte s hodnotou 1 alebo 0. Pre ich ďalšie spracovanie funkciou decodeBaudot je ich však potrebné dostať do jedného bytu, kde každý bit bude zopovedať svojej pozícii v poli. Jednotlivé prvky poľa sú postupne pripočítavané k premennej, na ktorú je následne aplikovaný bitový posun doľava. Pripočítavanie začne vždy na prvku reprezentujúcom MSB a končí pri prvku reprezentujúcom LSB. Vznikne tak premenná, ktorej hodnota bude v dekadickej sústave zodpovedať číslu v rozsahu 0 až 31.

Číslo v tejto forme je ďalej predané funkcii decodeBaudot. Tá zabezpečuje prevod čísla z jeho binárnej formy do znaku, ktorý reprezentuje v Baudotovom kóde. Pozostáva z dvoch polí, jednej pre písmená a druhej pre znaky. Veľkosť týchto polí je nastavená na 2^5, čo zodpovedá počtu všetkých možných znakov v Baudotovom kóde. Prvky sú pritom zoradené podľa hodnoty ich binárnej reprezentácie takže na pozície 0 bude znak null ktorému zodpovedá v Baudotovom kóde 00000 ďalšie bude T ktorému zodpovedá 00001 a tak ďalej. Táto funkcia vráti symbol z príslušneho poľa, z pozície určenej hodnotou, ktorá je mu predaná v argumente. To z ktorej tabuľky bude číslo vybrané určuje príznak isLetters, v podobe globálnej premennej, ktorý sa nastavuje v prípade že dekódovaný kód zodpovedá symbolom FS alebo LS.

6.4 Forma dát

Aby bolo možné s dekódovanými dátami pracovať a zobraziť ich na displej je potrebné ich nejakým spôsobom ukladať do pamäte. Bolo na to zvolené pole stringov, ktorých dĺžka zodpovedá počtu znakov v riadku displeja. Ukladanie dát v tejto forme značne zjedndušuje ich zobrazenie na displeji. Množstvo Stringov nachádajúcich sa v poli je možné nastaviť premennou v programe. Predbežne bola zvolená hodnota 8, čo zodpovedá 8 riadkom textu na displeji a teda dvom plným obrazovkám. Spolu s týmto poľom sú definované aj dve premenné, firstRow popisujúci ktorý string reprezentuje začiatok textu a curentRow hovoriaci do ktorého stringu sa momentálne zapisujú novo prijaté dáta. V prípade že dôjde k prijatiu väčšieho množstva dát môže dôjsť k naplneniu všetkých stringov. V tejto sitúacii sa string na ktorý ukazuje premenná firstRow vyprázdni. Premenná firstRow sa nastaví tak aby ukazovala na nasledujúci reťazec a hodnota currentRow bude ukazovať na práve vyprázdnený reťazec. Týmto spôsobom bude dochádzať k cyklovaniu medzi týmito reťazcami a zariadenie bude môcť prijímať nové dáta do nekonečna. Užívateľ bude mať taktiež prístup k časti už prijatých dát, ktoré už niesú na obrazovke.

6.5 Užívateľské rozhranie

Ako rozhranie pre zobrazovanie dát pre užívateľa je použitý štvorriadkový displej, ktorý je k procesoru pripojený pomocou zbernice I2C. Komunikácia s displejom je zabezpečená pomocou knižnice LiquidCrystal_I2C, ktorá zabezpečuje väčšinu potrebnej réžie. Použitý displej ponúka tiež možnosť vytvorenia vlastných znakov. Táto funkcia bola použitá na vytvorenie niekoľkých znakov a šípiek, použitých v hornej lište v menu.

Užívateľ môže do behu programu zasahovať pomocou štyroch tlačidiel. Tlačidlá sú pomenované Up, Down, Left a Right. Užívateľské rozhranie sa skladá z troch sekcií:

- Zobrazovanie textu,
- Hlavné menu,
- Podmenu.

6.5.1 Zobrazovanie textu

Sekcia zobrazovanie textu slúži na zobrazenie textu, ktorý zariadenie zdekódovalo z prijatého signálu. Predstavuje úvodnú obrazovku ktorú užívateľ uvidí po spustení zariadenia. Táto sekcia sa môže zobrazovať v dvoch módoch. V neaktívnom móde sú všetky riadky obrazovky vyhradené pre zobrazovanie prijatých dát. Ak užívateľ interagoval so zariadením pomocou tlačidiel prepne sa zariadenie do aktívneho módu. Ak je zariadenie v aktívnom móde, na prvom riadku displeja sa bude zobrazovať horná lišta, ktorá obsahuje popis funkcií jednotlivých tlačidiel. V prípade že užívateľ po dobu 5 sekúnd nestlačil žiadne tlačdilo zariadenie sa opäť prepne do neaktívneho módu, kde sa všetky riadky displeja používajú pre zobrazenie dát.

Po spustení je zariadenie v móde automatického posuvu. Znamená to že sa na displeji budú zobrazovať dáta, ktoré sú aktuálne prijímané.

Displej, ktorý je v zariadení použitý umožňuje zobraziť maximálne 80 znakov v neaktívnom a 60 znakov v aktívnom móde. Počet znakov v pamäti je však vyšší. Užívateľ sa môže v prijatých dátach pohybovať pomocou tlačidiel Up a Down. Pohyb je pritom obmedzený prvým a posledným riadkom dát.

Stlačením tlačidla Up alebo Down sa taktiež preruší mód automatického posuvu a na displeji zostane zobrazený práve zvolený riadok. Ak sa chce užívateľ do režimu automatického posuvu vrátiť musí stlačiť tlačidlo Left.

Stlačenie tlačidla Right otvorí sekciu Hlavné menu.

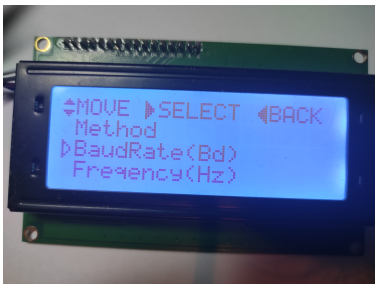
Obr. 6.1: Zobrazovanie textu

Obr. 6.2: Zobrazovanie textu s hornou lištou

6.5.2 Hlavné Menu

Sekcia Hlavné menu umožňuje užívateľovi zvoliť si, ktorý parameter chce nastavovať. Sú mu zobrazené všetky nastaviteľné parametre, medzi ktorými môže listovať pomocou tlačidiel Up a Down. Prvý riadok je opäť použitý pre zobrazenie funkcie tlačidiel. Ostatné tri riadky slúžia na zobrazenie jednotlivých položiek menu. Menu je cyklické, čo znamená že ak užívateľ listuje dostatočne dlho jedným smerom dostane sa späť na pôvodnú pozíciu. Práve zvolená položka je označená symbolom prázdnej šípky v prvom stĺpci displeja a nachádza sa vždy v treťom riadku.

Ak užívateľ stlačí tlačidlo Left, vráti sa späť na sekciu zobrazovanie textu. V prípade že stlačí tlačidlo Right, dostane sa do Podmenu, prislúchajúcemu práve zvolenej položke hlavného menu.



Obr. 6.3: Obrazovka hlavného menu

6.5.3 Podmenu

V sekcii Podmenu si užívateľ môže zvoliť spomedzi preddefinovaných hodnôt nastaviteľných parametrov. Podobne ako Hlavné menu, je aj toto menu cyklické a prvý riadok je rezervovaný pre zobrazenie funkcií tlačidiel. Hodnota, ktorú ma užívateľ práve zvolenú je označená prázdnou šípkou. Hodnota, ktorá je práve nastavená v

zariadení je naopak označená plnou šípkou. Medzi hodnotami je možné listovať pomocou tlačidiel Up a Down. Ak chce užívateľ nejaku z hodnôt aplikovať, musí ju zvoliť tak aby na ňu ukazovala prázdna šípka, a stlačiť tlačidlo Right. Úspešne zvolenie spozná tak, že sa predtým prázdna šípka zmení na plnú. Stalčením tlačidla Left sa užívateľ vráti späť do sekcie Hlavné Menu.

Obr. 6.4: Obrazovka podmenu

6.5.4 Obsluha tlačidiel

Obsluha tlačidiel prebieha v hlavnej slučke programu. V 5 ms intervaloch je kontrolované či je tlačidlo stlačené. Každému tlačidlu je pritom priradená premenná, ktorá sa pri každom cykle, počas, ktorého je tlačidlo stlačené inkrementuje. Ak nejaké z týchto počítadiel dosiahne hodnotu 10, čo znamená že tlačidlo bolo stlačené kontinuálne aspoň 50 ms, bude toto tlačidlo považované za stlačené. Bude tiež nastavaný príslušný príznak, buttonPressed, ktorý hovorí že je tlačidlo stlačené. Dochádza tak k eliminácii detekcie stlačenia tlačidla pri rôznych záchvevoch signálu.

Na základe nastavenia príznaku bude následne vykonaná akcia, ktorá bola danému tlačidlu priradená. Po vykonaní sa príznak buttonPressed nastaví na hodnotu false a príznak buttonProcessed na hodntu true. Dokým je príznak buttonProcessed v hodnote true, akcie tlačidla sú blokované. Jeho hodnota sa zmení na false až keď užívateľ stlačené tlačidlo pustí. Zabráni sa tak opakovanému vykonaniu priradenej akcie pri jednom stlačení.

6.6 Komunikácia s PC

Komunikácia medzi počítačom a zariadením využíva rozhranie USB. V kóde je teda najskôr potrebné nastaviť rýchlosť tejto sériovej linky a spustiť ju. K sériovej linke je priradená udalosť serialEvent, ktorá vyvolá prerušenie vždy keď sú na porte prijaté dáta.

Pre komunikáciu bol vytvorený jednoduchý protokol popísaný v časti 5. Zariadenie po pripojení k PC aplikácii zastáva funkciu dekodéru a jeho ovládacie prvky

sú deaktivované. Zabráni sa tak konfliktu pri jeho nastavovaní na počítači a v menu zároveň. Zariadenie sa do tohoto módu prepne po prijatí príkazu @+INIT. Dekódované dáta sú v tomto móde automaticky posielané na sériovú linku. Keďže na rozdiel od aplikácie na PC nedokáže mikrokontrolér jednoducho overovať či je sériová linka naďalej pripojená k počítaču so zapnutou aplikáciou, mohlo by dôjsť k zaseknutiu sa v tomto móde a potrebe manuálneho reštartu. Z tohoto dôvodu si zariadenie pri každom prijatom príkaze alebo odpovedi od PC aplikácie ukladá čas jej prijatsia. V hlavnej slučke je v pravidelných periódach kontrolované či čas od prijatia neprekročil definovaný interval. V prípade prekročenia vyšle mikrokontrolér správu v tvare @+TRY, na ktorú by mala aplikácia odpovedať. Každé vyslanie inkrementuje počítadlo. Ak dosiahne hodnotu 6, čo znamená že po dobu 3 sekundy neprijalo zariadenie žiadnu komunikáciu od PC aplikácie, zariadenie sa vráti späť do bežnej funkcie a komunikáciu je potrebné opätovne nadviazať.

Zariadenie ponúka tiež možnosť užívateľovi zapísať jednu hodnotu frekvencie do nevolatilnej pamäte. Táto frekvencia zostane v zariadení aj po jeho reštarte. Je na to využitá knižnica EEPROM.h. Táto knižnica pri použití na štandardných doskách arduino využívajúcich mikrokontroléry z rady atmega umožňuje zápis do ich pamäte EEPROM. Keďže nami použitý mikrokontrolér od spoločnosti STM pamäť EEPROM neobsahuje, využíva táto funkcia programovú pamäť flash. Táto pamäť sa každým zápisom opotrebováva. Počíta sa však s tým, že užívateľ túto hodnotu bude meniť len zriedkavo.

7 Program pre PC

Súčasťou tejto práce je program spustitelný na počítači, ktorý má umožňovať dodatočné nastavovanie parametrov zariadenia. Táto aplikácia by mala taktiež zobrazovať zariadením dekódované dáta. Má byť realizovaná pomocou užívateľského rozhrania a má byť napísaná v jazyku C#.

7.1 Framework .Net

7.1.1 Jazyk C#

C# je objektovo-orientovaný programovací jazyk vyvinutý spoločnosťou Microsoft. Je súčasťou frameworku .NET, ktorý predstavuje súbor technológií pre vývoj softwéru. Umožňuje vývoj webových, mobilných aj desktopových aplikácií. Zahŕňa tiež prostriedky pre tvorbu užívateľských rozhraní pre tieto aplikácie. Programy C# pozostávaju z jedného alebo viacerých súborov, pričom každý súbor môže obsahovať žiaden alebo viacero menových priestorov (namespace). Každý z týchto menových priestorov môže obsahovať ďalšie typy ako triedy, štruktúry rozhrania a iné [25]. Tento jazyk má nasledujúce vlastnosti:

- V jazyku C# neexistuje viacnásobna dedičnosť.
- Neexistujú globálne premenné. Každá premenná a metóda musí byť súčasťou triedy. Je však možné použiť statické metódy a verejné triedy.
- Je citlivý na veľké a malé písmená (Case sensitive).

7.1.2 WPF

WPF predstavuje open-source grafický subsystém, podobný staršiemu WinForms. Je súčasťou frameworku .NET a bol vytvorený za účelom tvorby užívateľských rozhraní pre aplikácie bežiace na operačnom systéme Windows. Pre svoju funkciu využíva DirectX. Jeho cieľom je oddeliť užívateľské rozhranie od logiky programu.

WPF je založené na jazyku značkovacom jazyku XAML (Extensible Application Markup Language), ktorý je podobný jazyku XML. Podobne ako XML je jeho úlohou určiť hierarchické rozloženie a vlastnosti jednotlivých objektov [26].

7.2 Realizácia programu

Program je rozdelený do dvoch súborov. Jeden zabezpečujúci funkciu programu a druhý obsahujúci definície užívateľského rozhrania.

7.2.1 Prepojenie s rozhraním

Aby bolo možné užívateľské rozhranie spraviť interaktívne, bolo potrebné ho nejakým spôsobom prepojiť s kódom vykonávajúcim ostatné funkcie. Na prepojenie tlačidiel a horného menu boli využté udalosti vyvolávané pri interakcii užívateľa s rozhraním. Tieto udalosti môžu mať rôzne typy a informujú program čo užívatel s komponentom rozhrania práve robí. Stlačenie tlačidla v užívateľskom rozhraní spôsobí vyvolanie udalosti click pre toto konkrétne tlačidlo. Vytvorením handleru, ktorý bude na túto udalosť reagovať, je možné tlačidlu pridať funkcionalitu.

7.2.2 Obsluha sériového portu

Komunikácia mezi zariadením a počítačom bude založená na sériovej komunikácii a použití rozhrania USB. Na obsluhu a komunikáciu po sériovej linke je možné v jazyku C# použiť triedu SerialPort. Táto trieda je súčasťou menného pristoru System.IO.Ports, a obsahuje metódy, ktoré umožňujú nastavenie parametrov sériovej komunikácie, výber použitého portu a prijímanie a odosielanie dát. Ponúka možnosti na detekciu prijatej komunikácie pomocou udalostí, ktoré sa vykonajú vždy po naplnení bufferu sériovej linky. Je vďaka nej možné detekovať aktívne Com porty počítača a zjednodušiť tak jeho voľbu.

Aplikácia najskôr čaká na otvorenie sériového portu užívateľom. Pri otvárani portu je potrebné sledovať či je zvolený správny názov portu, či tento port nieje už otvorený iným zariadením alebo či tento port vôbec existuje. Je taktiež potrebné zvoliť správne parametre komunikácie. Táto aplikácia rovnako ako zariadenie používa rýchlosť 9600 Bd. Po jeho otvorení sa k nemu priradí handler, ktorý reaguje na udalosť DataRecieved. Tá sa vyskytne po pri prijatí sekvencie dát alebo po prijatí znaku EOF. Keďže vyvolanie tejto udalosti určuje priamo operačný systém, nieje zaručené že sa zavolá po prijatí každého bytu dát. Po prijatí sú dáta skontrolované. ak obsahujú @, ktorý indikuje že reťazec obsahuje správu od zariadenia. Ak sa v texte príkaz nájde, je z neho vyňatý a ostatné dáta sa zobrazia v užívateľskom rozhraní.

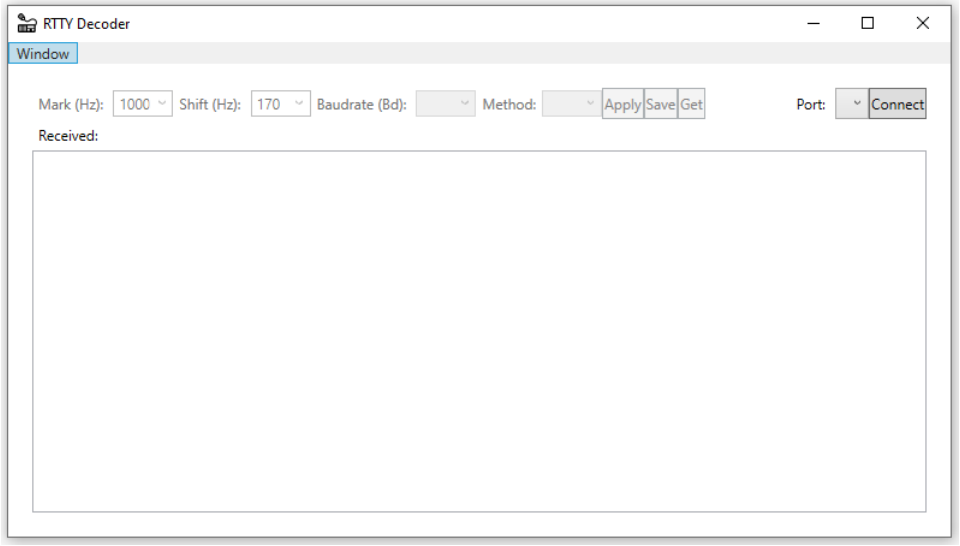
Program sa pri pripájaní k zariadeniu drží postupu zobrazeného na 5.1. Po otvorení portu teda vyšle správu @+INIT. V prípade že zariadenie pripojené na tomto porte na správu neodpovie, pripojenie sa bude považovať za neúspešné. Príchod odpovede sa pritom kontroluje časovačom, ktorý sa spustí po otvorení portu a jeho dĺžka je nastavená na 1 s. Ak príde odpoveď, tak sa na základe prijatých dát nastavia príslušné premenné zodpovedajúce hodnotám parametrov v užívateľskom rozhraní. Je pritom kontrolované či parametre majú formu číselnej hodnoty.

Po nadviazaní spojenia program vypisuje prijaté dáta na obrazovku a čaká na interakciu od uživateľa alebo od zariadenia. V stlačenia tlačidla v rozhraní ja zavolaná príslušná funkcia

Užívateľ taktiež môže prijatý text uložiť do súboru na čo je využitá trieda SaveFileDialog. Táto trieda zabezpečuje otvorenie okna pre uloženie súboru. Užívateľ v tomto okne môže nastaviť názov súboru a zvoliť lokáciu kam má byť uložený. Je tiež možné špecifikovať, akú príponu má tento súbor mať parametrom Filter.

7.2.3 Uživateľské rozhranie

Užívateľské rozhranie bolo realizované pomocou WPF a jazyja XAML. Pri tvorbe rozhrania bola snaha o zachovanie škálovatelnosti a určitej štruktúry a funkčného rozdelenia prvkov. Poloha jednotlivých komponent nieje definovaná pevne ale pomocou prvkov grid a panel. Grid umožňuje rozdelenie okna na na mriežku, veľkosť ktorej môže byť prispôsobená veľkosti prvku nachádajúceho sa v nej alebo prispôsobená vzhľadom k ostatným prvkom gridu. StackPanel zas vykresluje prvky, ktoré obsahuje zľava doprava resp. zhora nadol. Pre vytvorenie hornej lišty v aplikácii bol tiež použitý prvok DockPanel, ktorý definuje umiestnenie komopnent vľavo, vpravo, hore alebo dole. Použitie týchto prvkov by malo zabezpečiť že užívateľ môže ľubovoľne meniť veľkosť jej okna a štruktúra a funkčnosť rozhrania zostane zachovaná.



Obr. 7.1: Okno aplikácie

Aplikácia sa skladá z hlavného okna. To je rozdelené na časť riadiacich prvkov a časť pre zobrazovanie prijatých dát, a hornej lišty menu.

Lišta menu sa nachádza v hornej časti obrazovky a obsahuje jedno podmenu názvom window. Toto podmenu umožňuje užívateľovi uložiť prijatý text do súboru a vyčistit pole pre zobrazovanie textu. Nachádza sa tu aj možnosť ukončiť aplikáciu.

Pod týmto menu sa nachádza riadok obsahujúci prvky pre nastavenie jednotlivých parametrov signálu. Táto sekcia je po spustení zariadenia deaktivované pomo-

cou parametra isEnable stackPanelu v ktorom sa tieto prvky nachádzajú. Sprístupní sa až po pripojení aplikácie k zariadeniu. Veľkosť jednotivý parametrov je možné nastavovať pomocou ComboBoxov. ComboBoxy pre veličiny frekvenčný posuv Shift a frekvenciu Mark sú pritom editovateľné a umožňujú užívateľovi vložiť vlastnú číselnú hodnotu. Nastavenia je následne možné tlačidlom Apply poslať do zariadenia. Tlačidlo Save zase umožňuje zvolenú frekvenciu zapísať do nevolatilnej pamäťe zariadenia. Pomocou tlačidla Get je zas možné získať aktuálne nastavenie zariadenia.

Vpravo od nastavenia parametrov sa nachádza ComboBox pre voľbu portu, na ktorý je pripojené zariaenia a tlačidlo pre pripojenie. Po stlačení tlačidla a pripojení k zariadeniu môže toto tlačidlo byť použité aj na odpojenie.

8 Merania

V rámci tejto práce bolo realizovaných niekoľko rôznych metód demodulujúcich FSK signál používaný pri prenose RTTY. Keďže je RTTY prenášané rádiovým kanálom, bude spolu so signálom prijímané rušenie. Toto rušenie môže mať vplyv na rozpoznateľnosť jednotlivých frekvencií v signále a môže tým zvýšiť chybovosť pri dekódovaní. Je preto potrebné otestovať jednotlivé implementované metódy a zistiť, či sú schopné fungovať aj pri zarušenom signále.

8.0.1 Rušenie

Rušenie predstavuje nežiadúce frekvencie, nachádzajúce sa v signále. Zdrojmi rušenia pri rádiovom prenose môžu byť prírodné elektromagnetické procesy ako aj iné rádiové vysielače. Je teda nevyhnutnou súčasťou pri prenose rádiovými kanálmi. Rušenie je možné odstrániť prefiltrovaním signálu tak, aby v ňom zostali len žiadané zložky. Keďže naše zariadenie demoduluje signál softvérovo a nesústredí sa na konkrétny frekvenčný pár nebolo možné signál takto predfiltrovať.

Rušenie v rádiovom kanáli bude simulované pomocou zavedenia bieleho šumu do signálu. Pridanie bieleho šumu bude realizované pomocou programu Audacity. Tento program umožňuje nahranie signálu z výstupu zvukovej karty pomocou funkcie loopback. Umožňuje taktiež jednoduché pridanie šumu so zvolenou úrovňou do signálu.

8.0.2 Generátory RTTY signálu

Aby bolo možné zariadenie otestovať a realizovať merania, bolo nutné nájsť vhodný spôsob generovania signálu. Je možné použiť hardvérové riešenie v podobe digitálneho modemu. Z hľadiska dostupnosti je však dnes medzi rádioamatérmi, z dôvodu dostupnosti a ceny, populárnejším riešením využitie softvéru bežiaceho na počítači alebo mobile. Na internete je možné nájsť množstvo rôznych aplikácií, s rôznymi možnosťami nastavenia rozsahu frekvencií, rýchlostí prenosu a inými. Veľká časť týchto aplikácií však nieje kompatibilná s modernými operačnými systémami, alebo nieje dostupná z dôvodu expirácie linku na stiahnutie. Podarilo sa však nájsť niekoľko funkčných generátorov.

DroidRTTY

DroidRTTY je aplikácia umožňujúca dekódovať a generovať RTTY signál bežiaca na mobilnom operačnom systéme Android. Umožňuje prácu s frekvenciou v rozsahu od

160 do 3000 Hz. Ponúka tri nastavenia symbolovej rýchlosti a frekvenčného posunu a to [27]

- 45,4 Bd a 170 Hz
- 50 Bd a 450 Hz
- 50 Bd a 85 Hz

TrueTTY

Prvou z nich bola aplikácia TrueTTY. Je ju možné nájsť na [28] Táto aplikácia umožňuje okrem RTTY prácu aj s AMTOR, BPSK a niekoľko ďalších typov komunikácie. Dokáže fungovať ako dekodér a zároveň ako generátor signálu. Ako vstup pritom využíva vstup zvukovej karty počítača, Vygenerovaný signál naopak odosiela na zvukový výstup. Aplikácia tiež má možnosť automatickej detekcie frekvencie prijímanej komunikácie.

Je v nej možné nastaviť nasledovné parametre RTTY komunikácie

- Frekvenčný posun: 23 až 1200 Hz
- Rýchlosť prenosu: 25 až 1200 Bd,
- Mark frekvenciu: 300 až 4000 Hz v závislosti na použitej hodnote frekvenčného posuvu.

Je taktiež možné nastaviť, či má byť vysielaný signál invertovaný. Neumožňuje bohužial nastavenie počtu použitých stop bitov v kóde [28]. Program používa 1,5 stop bitu, pri meraní na osciloskope však nebola dĺžka tohoto stopbitu konzistentná, čo by mohlo mať negatívny dopad na dekódovanie a demoduláciu signálu.

8.0.3 MMTTY

Ďalšou aplikáciou, fungujúcou na Windows 10 je MMTTY, ktorá je dostupná na [29]. Táto aplikácia je určená špecificky pre prácu so signálom RTTY. Rovnako ako aplikácia TrueTTy využíva pre svoju funkciu zvukovú kartu počítača.

Je v ňom možné nastaviť nasledovné parametre RTTY komunikácie

- Frekvenčný posun: 23 až 850 Hz
- Rýchlosť prenosu: 20 až 600 Bd,
- Mark frekvenciu: 915 až 2125 Hz

Existuje tiež možnost prepínania invertovaného signálu. Signál generovaný týmto softvérom využíva 2 stop bity. Naviac tento program ponúka viacero metód demodulácie signálu [29].

Z dôvodu jednoduchšieho ovládania, širších možností nastavenia a vyššej konzistentnosti signálu bol pre merania zvolený práve tento program.

8.0.4 Osciloskop

Na zobrazenie meraného signálu bol použitý osciloskop typu PICOSCOPE2205A MSO. Tento osciloskop umožňuje pripojenie dvoch kanálov. Má taktiež možnosť zobraziť frekvenčné spektrum meraného signálu. Po pripojení k PC je možné merané dáta sledovať priamo v aplikácii. Bola využitá funkcia uloženia meraných dát. Tieto dáta možno otvoriť v picoscope aplikácií a ďalej ich spracovať, prípadne ďalej exportovať do formy obrázku alebo .csv súboru [30].

8.1 Postup merania

Pre potreby merania bude najskôr vygenerovaný RTTY signál. Tento signál bude obsahovať správu . Ten bude uložený vo forme audio súboru, čím sa predíde možnosti vzniku nepresnosti spôsobenej variabilitou generátora. Následne budú vytvorené verzie tohoto signálu s rôznymi úrovňami rušenia v podobe bieleho šumu.

Na meranie bude teda použitých 6 rôznych signálov. Signál bez rušenia, na ktorom bude možné overiť funkciu jednotlivých metód a namerané hodnoty budú slúžiť ako základ pre porovnanie. Následne budú použité signály s úrovňou bieleho šumu $0,2$, $0,4$, $0,6$,$0,8$ a 1.

Tieto signály budú postupne privádzané do zariadenia a pomocou jednotlivých metód dekódované. Na sériový port budú posielané vždy po demodulovaní jednotlivé bity a dekódované znaky. Na základe počtu správne dekódovaných znakov bude vyhodnotená chybovosť ako

$$CER = \frac{N_{Cerr}}{N_C} \tag{8.1}$$

kde N_{Cerr} označuje počet nesprávne dekódovaných znakov a N_C počet všetkých znakov. Meranie bude prebiehať na RTTY frekvencii 1000 Hz s frekvenčným posuvom 170 Hz a symbolovou rýchlosťou 50 Bd. Bude taktiež vyhodnocovaná chybovosť BER, ktorú vypočítame ako

$$BER = \frac{N_{berr}}{N_b} \tag{8.2}$$

kde N_{berr} označuje počet chybných bitov a N_B počet všetkých odoslaných bitov. Výsledky budu spracované do tabuliek.

Pomocou osciloskopu budú tiež zobrazené priebehy na vstupe PA0 a PA1, kde bude možné sledovať vplyv zarušeného signálu.

8.2 Výsledky merania

Úroveň rušenia[%]	N_B	N_{Berr}	BER [%]	N_C	N_{Cerr}	CER [%]
0	328	0	0	40	0	0
20	328	0	0	40	0	0
40	328	0	0	40	0	0
60	328	18	5,487805	40	12	30
80	328	25	7,621951	40	14	35
100	328	46	14,02439	40	27	67,5

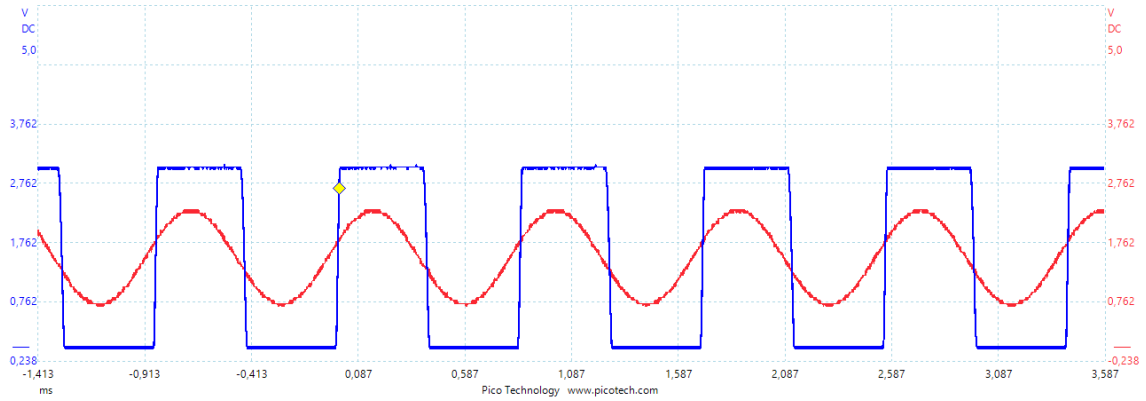
Tab. 8.1: Meranie metódou FFT

Úroveň rušenia [%]	N_B	N_{Berr}	BER [%]	N_C	N_{Cerr}	CER [%]
0	328	0	0	40	0	0
20	328	4	1,219512	40	3	7,5
40	328	42	12,80488	40	17	42,5
60	328	328	100	40	40	100
80	328	328	100	40	40	100
100	328	328	100	40	40	100

Tab. 8.2: Meranie metódou ISR

Úroveň rušenia [%]	N_B	N_{Berr}	BER [%]	N_C	N_{Cerr}	CER [%]
0	328	0	0	40	0	0
20	328	28	8,536585	40	13	32,5
40	328	212	64,63415	40	38	95
60	328	325	99,08537	40	40	100
80	328	328	100	40	40	100
100	328	328	100	40	40	100

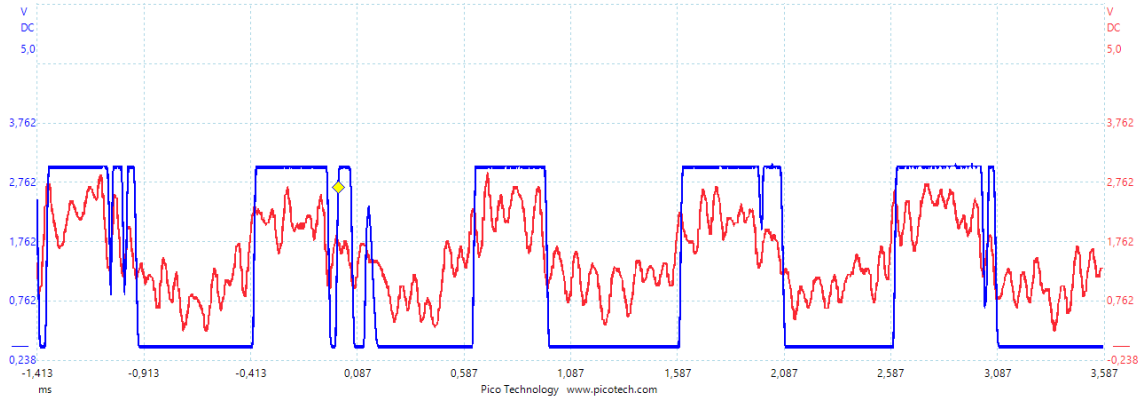
Tab. 8.3: Meranie metódou PulseIn



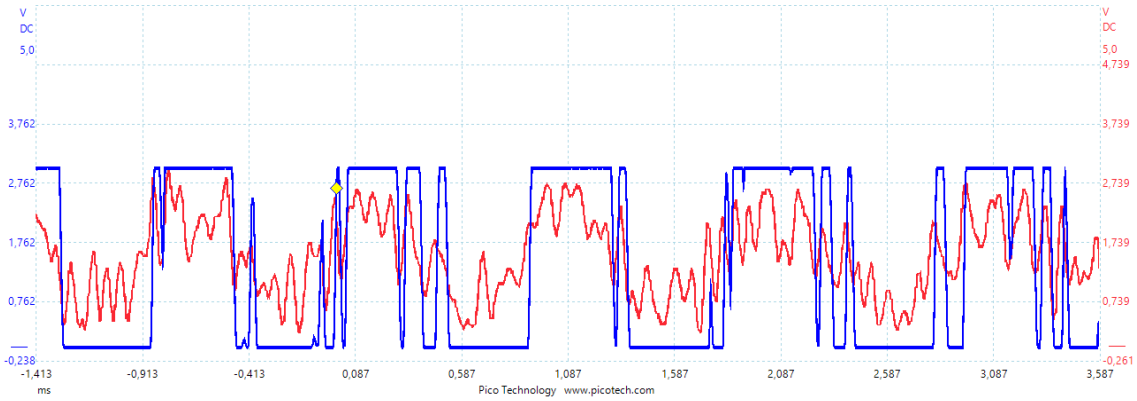
(a) Signál bez rušenia

(b) Signál so šumom 20%

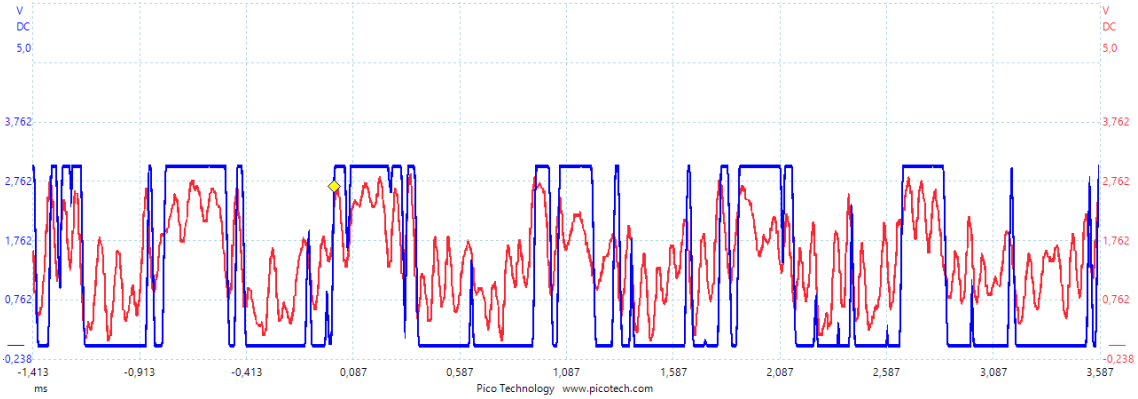
(c) Signál so šumom 40%



(d) Signál so šumom 60%

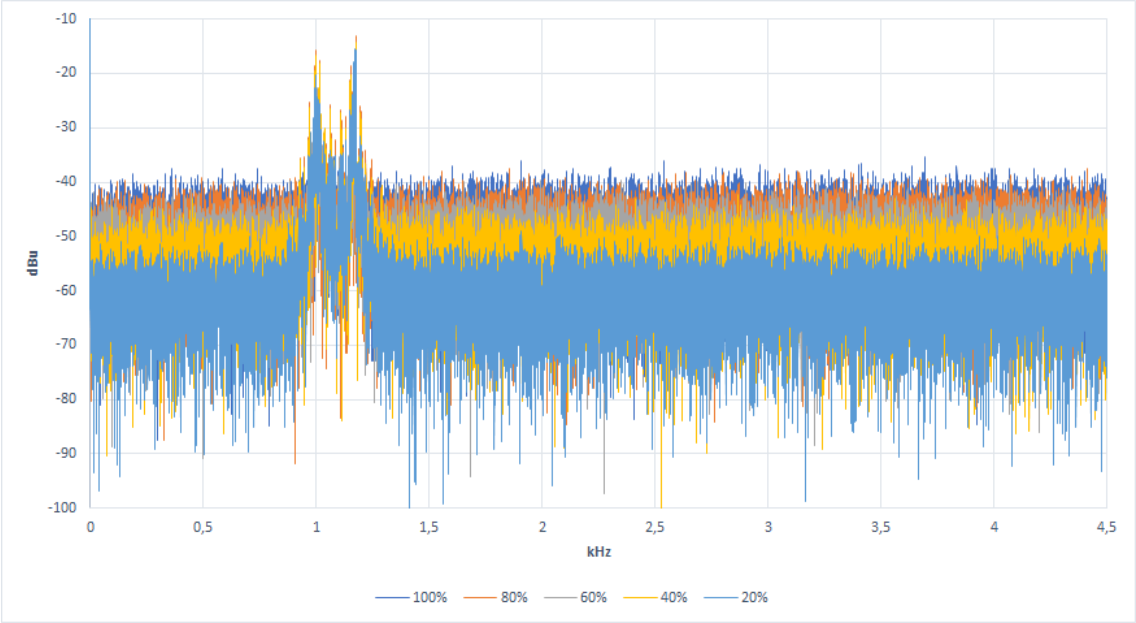


(e) Signál so šumom 80%



(f) Signál so šumom 100%

Obr. 8.0: Priebehy signálu na vstupoch PA1(modrá) a PA0(červená)



Obr. 8.1: Spektrum signálov so šumom

8.3 Vyhodnotenie

Merania na zariadení boli zrealizované. Získané hodnoty boli spracované do tabuliek. Boli tiež uložené priebehy signálu pri rôznych úrovniach pridaného šumu. Spektrá signálu pri rôznych úrovniach rušenia boli uložené a pomocou aplikácie vyexportované do formátu .csv. Následne z nich bol vytvorený graf.

Prvá bola testovaná metóda využívajúca FFT. Táto meóda dokázala bezchybne dekódovať signály so šumom 40%. Pri vyšších úrovniach šumu už začínalo dochádzať k vzniku chýb. Signály so šumom 60 a 80% boli dekódované s chybovosťou CER 30 a 35%. Približne tretina znakov je teda stratená, ale z textu môže byť stále možné získať časť informácie. Pri úrovni rušenia 100% chybovosť stúpla až na 67,5%.

Metóda ISR bola schopná bezchybne dekódovať nezarušený signál. Pri signále s rušením 20% došlo k nesprávnemu dekódovaniu 3 znakov, čo zodpovedá hodnote CER 7,5%. Došlo pritom len k 4 chybám pri prijímaní bitov. Znamená to, že takmer každá chyba, ktorá pri demodulácii vznikne spôsobí stratu celého znaku. Pri signále s úrovňou rušenia sa podarilo úspešne zdekódovať približne dve tretiny znakov a hodnota CER bola 42,5%. Pri vyšších rušeniach táto metóda už nedokázala jednoznačne určit frekvenciu prijímaného signálu a nepodarilo sa pomocou nej dekódovať ani jeden bit.

Podobne ako pri predchádzajúcich metódach, nedošlo pri dekódovaní signálu metódou pulseIn k vzniku žiadnej chyby. Už pri signále s rušením 20% bolo 32,5% znakov prijatých s chybou. Zo signálu s rušením 40% sa podarilo dekódovať len 2 znaky. V ďalších signáloch sa už nepodarilo správne demodulovať ani jeden bit.

Z nameraných hodnôt je zrejmé, že metódou najodolnejšou voči rušeniu je metóda FFT. Je založená na analýze jednotlivých frekvenčných zložiek signálu. Na obrázku je vidieť frekvenčné spektrum jednotlivých signálov. Frekvencie nesúce RTTY signál sú aj pri najvyššom použitom rušení rozpoznateľné. Z tohoto dôvodu dokáže táto metóda detekovať tieto frekvencie aj v zarušenom signále. Zobrazené spektrum je však vytvorené pri zásadne vyššej vzorkovacej frekvencii a počte vzoriek. Keďže naša verzia FFT využíva z dôvodu časovej náročnosti len 64 vzoriek, je vyššia šanca, že pri detekovaní frekvencie dôjde k vzniku chyby. Presnosť by bolo teda možné zvýšiť použitím výkonnejšieho hardvéru, ktorý by dokázal v čase jednej periódy signáloveho prvku počítať FFT s väčším počtom vzorkov. Rýchlosť by mohla zvýšiť tiež implementácia Goertzelovho algoritmu, pomocou ktorého je možné počítať len hodnotu konkrétnej časti frekvenčného spektra.

Chybovosť metód ISR a pulseIn je vyššia. Je spôsobená nekonzistentným tvarom obdĺžnikového signálu, ktorý vzniká na komparátore. Tvar zarušeného signálu je rušením zdeformovaný. Táto deformácia spôsobuje, že hodnota tohoto signálu stúpne a klesne pod hodnotu komparačnej hladiny komparátoru. Tento jav je možné vidieť

na priebehoch v časti 8.0. Chybovosť týchto metód by bolo možné znížiť pripojením filtra na vstup signálu. Tento filter by zo signálu odstránil nežiadúce frekvencie. Použitím filtra by však zariadenie prišlo o možnosť fungovať na inom frekvenčnom páre ako na tom, ktorý filter prepúšťa.

Z meraní je taktiež možné vidieť, že aj malá chybovosť BER dokáže spôsobiť vysokú chybovosť znakov CER. Vyplýva to z vlastností Baudotovho kódu, ktorý je pri RTTY využívaný. Tento typ kódovanie neponúka žiadnu možnosť detekcie chyby ani opravy chyby, či už v podobe paritného bitu, CRC alebo iného. Z tohoto dôvodu spôsobí nesprávne prijatie jedného bitu, stratu celého znaku. Ak by išlo o znak LS alebo FS, mohlo by to spôsobiť prenesenie aj do ďalších znakov až do zaslania ďalšieho LS alebo FS.

Záver

Teoretická časť práce bola zameraná na popis rádiovej ďalekopisnej komunikácie a jej dekódovanie. V rámci prvej kapitoly bol popísaný historický vývoj RTTY a 5-bitových kódov, ktoré využíva a technológie ktoré využíva.

V nasledujúcej kapitole bola bližšie popísaná problematika klúčovania FSK a jeho demodulácie. Boli v nej popísané aj niektoré metódy demodulácie.

Na základe získaných poznatkov boli sformulované kritéria pre voľbu platformy. Najlepšie ich spĺňali mikrokontroléry z rodiny STM32. Pre túto aplikáciu bol zvolený konkrétne mikokontrolér STM32F411CE. Z dôvodu zjednodušenia realizácie plošného spoja bola zvolená varianta vývojového kitu Blackpill. Bol taktiež vytvorený návrh hardwéru zariadenia, bližšie rozobraný v kapitole. Tento návrh zahŕňa napájanie z batérie typu Li-Ion, ktorú umožňuje aj nabíjať. Prepojenie s inými zariadeniami a nabíjanie zariadenia, je najmä z dôvodu širokej kompatibility, zabezpečené rozhraním USB. Návrh disponuje tiež štyrmi tlačidlami pre jeho nastavovanie a štvorriadkovým displejom na zobrazenie dekódovaných výsledkov a menu.

Schéma návrhu bola vytvorená v programovom prostredí Eagle a na jej základe bol zrealizovný návrh plošného spoja. Tento plošný spoj bol osadený a sprevádzkovaný. Pri sprevádzkovaní bola postupne overovaná funkčnosť jeho častí a kontrolovali sa úrovne napätia.

Súčasťou tejto práce je aj aplikácia pre počítač, ktorá má s naším zariadením komunikovať. Bola preto vytvorená jednotná schéma tejto komunikácie v podobe protokolu. Ten zabezpečuje nadviazanie a ukončenie komunikácie, prípadne nastavenie dát zaraenia.

Pre osadené zariadenie bol následne vytvorený softvér zabezpečujúci demoduláciu FSK signálu a následné dekódovanie RTTY. Bolo taktiež vytvorené rozhranie pre displej, ktoré umožňuje nastaviť jednotlivé parametre signálu a zobrazovať dekódované dáta.

Aby bolo možné dekódované dáta zobraziť na počítači bola zrealizovaná aj počítačová aplikácia v jazyku C#. Pre túto aplikáciu bolo vytvorené rozhranie, ktoré umožňuje užívateľovi nastavovať parametre zariadenia a vidieť prijaté dáta.

Aby bola overená funkčnosť zariadenia aj pri zarušenom signále boli v rámci tejto práce vykonané merania. Výsledky týchto meraní boli spracované do tabuliek. Bol taktiež zobrazený priebeh signálu a jeho spektrum pri všetkých meraných úrovniach rušenia.

Výsledkom tejto práce je zariadenie, ktoré dokáže dekódovať rádioamatérsku komunikáciu využívajúcu technológiu RTTY. Zariadenie je autonómne a napájané z batérie, ktorú je možne nabíjať pomocou USB-C konektora. Softvérová realizácia demodulácie umožňuje dekódovanie aj pri použití neštandardných frekvenčných pá-

rov. Boli implementované tri metódy demodulácie. Dekódovanie pomocou týchto metód bolo otestované aj na zarušenom signále. Najodolnejšou metódou voči rušeniu sa ukázala byť metóda FFT. Metóda ISR dokázala dekódovať mierne zarušený signál, ale pri vyšších rušenia zlyhala. Chybovosť metódy pulseIn bola vysoká už pri najnižšej úrovni vneseného rušenia.

Zariadenie by bolo možné v budúcnosti rozšíriť o LED alebo OLED display, ktorý by širšie možnosti zobrazenia. Program mikrokontroléra by bolo možne zoptimalizovať prepísaním pomocou HAL funkcií. Bolo by tiež možné implementovať demoduláciu pomocou Goertzelovho algoritmu, ktorá by potenciálne mohla umožniť zariadeniu dekódovať aj signály s vyššími prenosovými rýchlosťami.

Literatúra

[1] Anton A. Huurdeman. *The Worldwide History of Telecommunications*. John Wiley & Sons, 2003.

[2] F. J. Singer. Military teletypewriter systems. *Electrical Engineering*, 68(1):39–39, 1949. doi:10.1109/EE.1949.6444554.

[3] Wayne Green. *Handbook*. G/L Tab Books, prvé edition, 1972.

[4] Alan Hobbs. Five-unit codes. URL: http://www.rtty.com/England/fiveunits.htm.

[5] Fuqin Xiong. *Digital modulation techniques*. Artech House, Boston, 2nd ed edition, 2006.

[6] Ing. Radim Číž Ph.D. *Principy modulací a přenosu sdělovacích signálů pro integrovanou výuku VUT*. Vysoké učení technické v Brně, Brno, první edition, 2014.

[7] Václav Žalud. *Moderní radioelektronika*. BEN - technická literatura, Praha, 1. edition, 2000.

[8] A. Ghazi, J. Boutellier, J. Hannuksela, S. Shahabuddin, and O. Silvén. Programmable implementation of zero-crossing demodulator on an application specific processor. In *SiPS 2013 Proceedings*, pages 231–236, 2013. doi:10.1109/SiPS.2013.6674510.

[9] Richard G. Lyons. *Understanding digital signal processing*. Prentice Hall, Upper Saddle River, 3rd ed edition, 2011.

[10] Michael Cerna and Audrey Harvey. The fundamentals of fft-based signal analysis and measurement, 2000. URL: https://www.sjsu.edu/people/burford.furman/docs/me120/FFT_tutorial_NI.pdf.

[11] C. Sidney Burrus. *Moderní radioelektronika*. Libretexts, 2020. URL: https://eng.libretexts.org/Bookshelves/Electrical_Engineering/Signal_Processing_and_Modeling/Book%3A_Fast_Fourier_Transforms_(Burrus).

[12] Microchip. *ATmega640/V-1280/V-1281/V-2560/V-2561/V Datasheet*, 2020. URL: https://ww1.microchip.com/downloads/en/DeviceDoc/ATmega640-1280-1281-2560-2561-Datasheet-DS40002211A.pdf.

[13] Microchip. *ATmega164A/PA/324A/PA/644A/PA/1284/P Datasheet*, 2020. URL: https://ww1.microchip.com/downloads/en/DeviceDoc/ATmega164A_PA-324A_PA-644A_PA-1284_P_Data-Sheet-40002070B.pdf.

[14] STMicroelectronics. *STM32F103x8, STM32F103xB Datasheet*, 2015. URL: https://www.st.com/resource/en/datasheet/stm32f103c8.pdf.

[15] STMicroelectronics. *STM32F411xC, STM32F411xE Datasheet*, 2017. URL: https://www.st.com/resource/en/datasheet/stm32f411ce.pdf.

[16] *Bluepill*. URL: https://stm32-base.org/boards/STM32F103C8T6-Blue-Pill.html.

[17] Blackpill. URL: https://stm32-base.org/boards/STM32F103C8T6-Black-Pill.html.

[18] Hunan Sounddon New Energy. *Lithium-ion Battery Product Specification*. URL: https://datasheetspdf.com/pdf-file/1100076/HunanSounddon/18650/1.

[19] NanJing Top Power ASIC. *TP4056 1A Standalone Linear Li-Ion Battery Charger with Thermal Regulation in SOP-8*. URL: https://dlnmh9ip6v2uc.cloudfront.net/datasheets/Prototyping/TP4056.pdf.

[20] Fortune Semiconductor Corp. *One Cell Lithium-ion/Polymer Battery Protection IC DW01-P*, 2006. URL: https://cdn.sparkfun.com/assets/learn_tutorials/2/5/1/DW01-P_DataSheet_V10.pdf.

[21] AEROSEMI. *MT3608 High Efficiency 1.2MHz 2A Step Up Converter*. URL: https://www.olimex.com/Products/Breadboarding/BB-PWR-3608/resources/MT3608.pdf.

[22] Arduino. URL: https://www.arduino.cc/en/guide/introduction.

[23] Arduino pulsein. URL: https://www.arduino.cc/reference/en/language/functions/advanced-io/pulsein/.

[24] Arduinofft. URL: https://github.com/kosme/arduinoFFT.

[25] Csharpcorner. URL: https://www.c-sharpcorner.com/article/what-is-c-sharp/.

[26] Wpf. URL: https://docs.microsoft.com/en-us/visualstudio/designers/getting-started-with-wpf?view=vs-2019.

[27] Droidrtty. URL: https://play.google.com/store/apps/details?id=com.wolphi.droidrtty&hl=en_US&gl=US.

[28] Truetty. URL: https://www.dxsoft.com/en/products/truetty/.

[29] Mmtty. URL: https://hamsoft.ca/pages/mmtty.php.

[30] Picoscope. URL: https://www.picotech.com/oscilloscope/2000/picoscope-2000-overview.

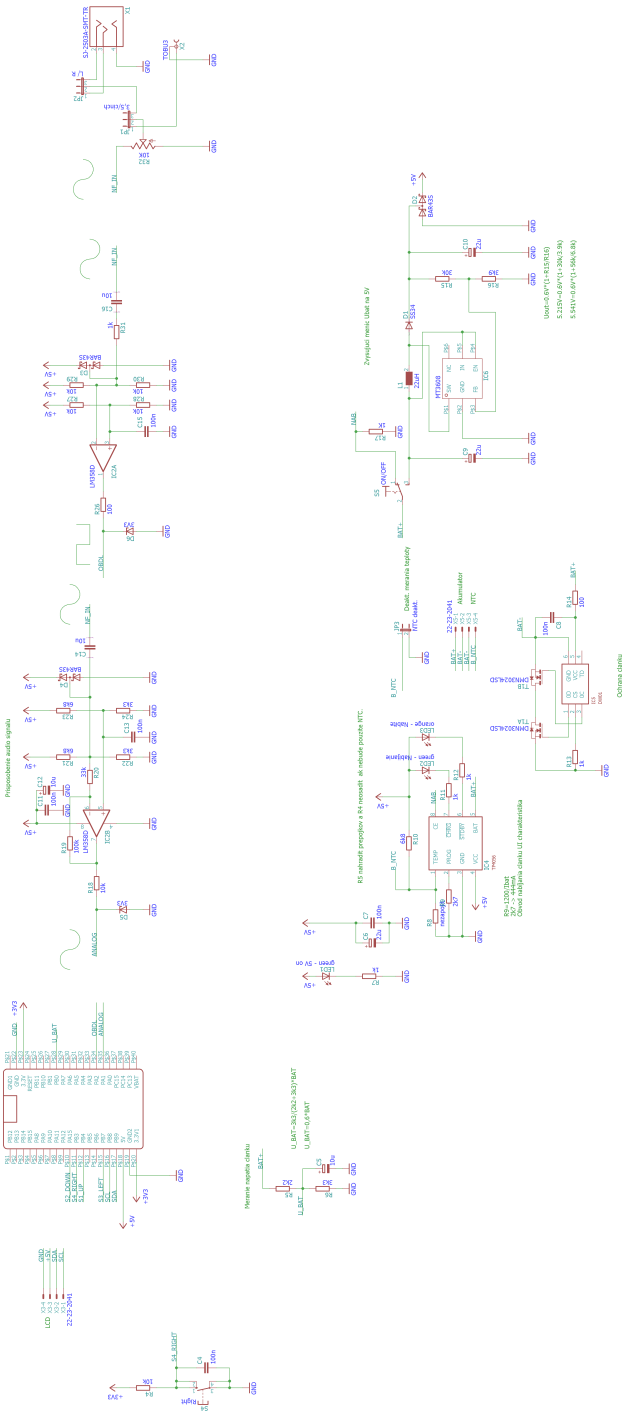
Zoznam symbolov, veličín a skratiek

ADC	Analog-to-Digital Converter – analógovo-digitálny prevodník
ARM	Advanced RISC Machine
BER	Bit Error Rate – bitová chybovosť
CER	Character Error Rate – znaková chybovosť
CR	Carriage Return – návrat vozíka
DFT	Digital Fourier Transform – digitálna Fourierova transformácia
DMA	Direct Memory Access – priamy prístup k pamäti
ESD	Electrostatic Discharge – elektrostatický výboj
FFT	Fast Fourier Transform – rýchla Fourierova transformácia
FM	Frekvenčná Modulácia
FS	Figure Shift – prepnutie symbolov
FSK	Frequency Shift Keying – kľúčovanie frekvenčným posunom
ISR	Interrupt Service Routine – obsluha prerušenia
ITA	International Telegraph Alphabet – medzinárodná telegrafná abeceda
LF	Loop Filter – filter slučky
LS	Letter Shift – prepnutie písmen
NTC	Negative Temperature Coefficient — negatívny teplotný koeficient
PLL	Phase Locked Loop — fázový záves
RTTY	Radio Teletype — rádioý ďalekopis
SMD	Surface Mount Device — súčiastky pre povrchovú montáž
SWD	Serial Wire Debug
UART	Universal asynchronous receiver-transmitter
USB	Universal Serial Bus
VCO	Voltage Controlled Oscillator – oscilátor riadený napätím

WPF Windows Presentation Foundation

Zoznam príloh

A Schéma zapojenia



Obr. A.1: Schéma zapojenia

B Zoznam súčiastok

Part	Value	Device	Package
C1	100n	C-EUC1206	C1206
C2	100n	C-EUC1206	C1206
C3	100n	C-EUC1206	C1206
C4	100n	C-EUC1206	C1206
C5	10u	CPOL-EUSMCC	SMC_C
C6	22u	CPOL-EUSMCC	SMC_C
C7	100n	C-EUC1206	C1206
C8	100n	C-EUC1206	C1206
C9	22u	CPOL-EUSMCC	SMC_C
C10	22u	CPOL-EUSMCC	SMC_C
C11	100n	C-EUC1206	C1206
C12	10u	CPOL-EUSMCC	SMC_C
C13	100n	C-EUC1206	C1206
C14	10u	CC1206	C1206
C15	100n	C-EUC1206	C1206
C16	10u	CC1206	C1206
D1	SS34	DIODE-DO214AC	DO214AC
D2	BAR43S	BAS40-04	SOT23
D3	BAR43S	BAS40-04	SOT23
D4	BAR43S	BAS40-04	SOT23
D5	3V3	ZENER-DIODESOD80C	SOD80C
D6	3V3	ZENER-DIODESOD80C	SOD80C
IC1	STM32F103C8T6	BLUE_PILL	BLUE_PILL
IC2	LM358D	LM358D	SO08
IC4	TP4056	TP4056	SOT23-8
IC5	DW01	DW01	SOT23-6
IC6	MT3608SOT23-6	MT3608SOT23-6	SOT23-6
JP1	3,5/cinch	JP2E	JP2
JP2	L/R	JP2E	JP2
JP3	NTC-deakt.	JP1E	JP1
L1	22uH	WE-PD-7447714680	WE-PD_1030/1050

Tab. B.1: Zoznam súčiastok 1/3

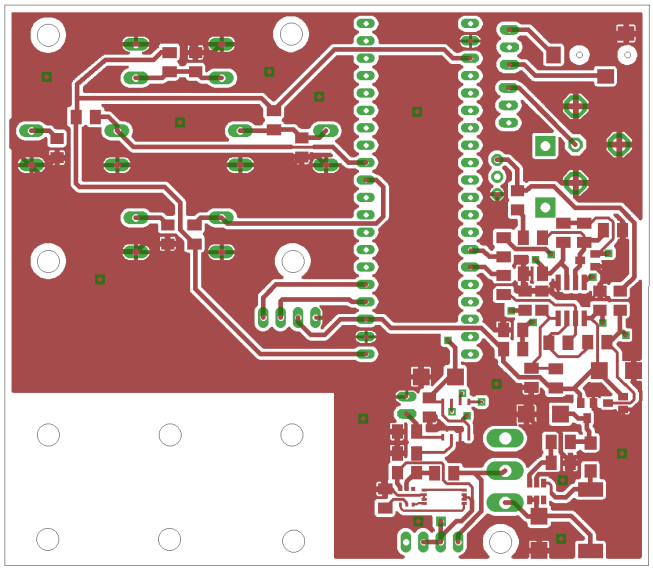
LED1	green-5Von	LEDCHIPLED_1206	CHIPLED_1206
LED2	green-Nabijanie	LEDCHIPLED_1206	CHIPLED_1206
LED3	orange-Nabite	LEDCHIPLED_1206	CHIPLED_1206
R1	10k	R-EU_M1206	M1206
R2	10k	R-EU_M1206	M1206
R3	10k	R-EU_M1206	M1206
R4	10k	R-EU_M1206	M1206
R5	2k2	R-EU_M1206	M1206
R6	3k3	R-EU_M1206	M1206
R7	1k	R-EU_M1206	M1206
R8	nezapojit	R-EU_M1206	M1206
R9	2k7	R-EU_M1206	M1206
R10	6k8	R-EU_M1206	M1206
R11	1k	R-EU_M1206	M1206
R12	1k	R-EU_M1206	M1206
R13	1k	R-EU_M1206	M1206
R14	100	R-EU_M1206	M1206
R15	30k	R-EU_M1206	M1206
R16	3k9	R-EU_M1206	M1206
R17	1K	R-EU_M1206	M1206
R18	10k	R-EU_M1206	M1206
R19	100k	R-EU_M1206	M1206
R20	33k	R-EU_M1206	M1206
R21	6k8	R-EU_M1206	M1206
R22	3k3	R-EU_M1206	M1206
R23	6k8	R-EU_M1206	M1206
R24	3k3	R-EU_M1206	M1206
R26	100	R-EU_M1206	M1206
R27	10k	R-EU_M1206	M1206
R28	10k	R-EU_M1206	M1206
R29	10k	R-EU_M1206	M1206
R30	10k	R-EU_M1206	M1206
R31	1k	R-EU_M1206	M1206
R32	10K	US-EVUF3M	EVUFXM

Tab. B.2: Zoznam súčiastok 2/3

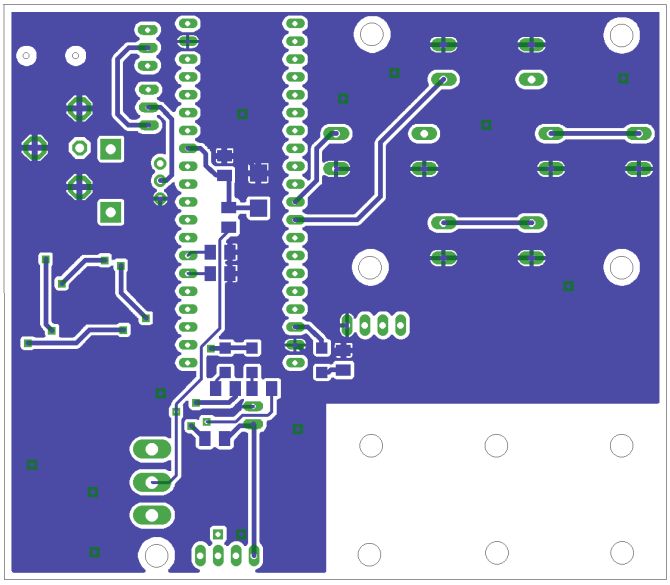
S1	Up	RACON12	RACON12
S2	Down	RACON12	RACON12
S3	Left	RACON12	RACON12
S4	Right	RACON12	RACON12
S5	ON/OFF	M9040P	M9040P
T1	DMN3024LSD	DMN3024LSD	TSSOP-8
X1	SJ-2503A-SMT-TR	SJ-2503A-SMT-TR	CUI_SJ-2503A-SMT-TR
X2	TOBU3	TOBU3	TOBU3
X3	22-23-2041	22-23-2041	22-23-2041
X5	22-23-2041	22-23-2041	22-23-2041

Tab. B.3: Zoznam súčiastok 3/3

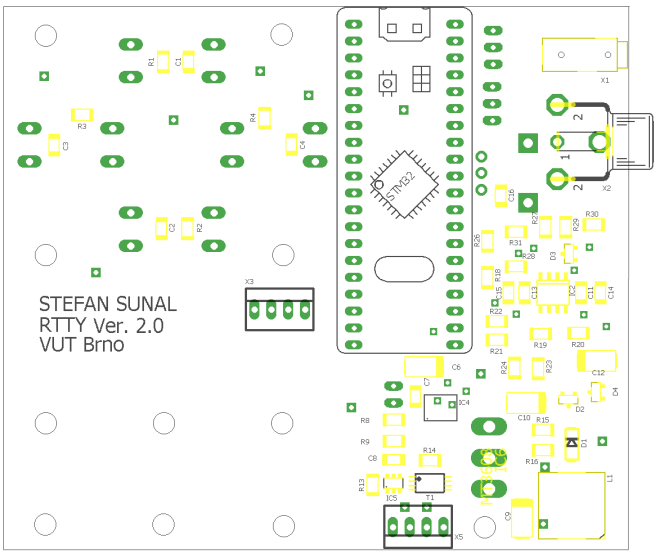
C Doska plošného spoja



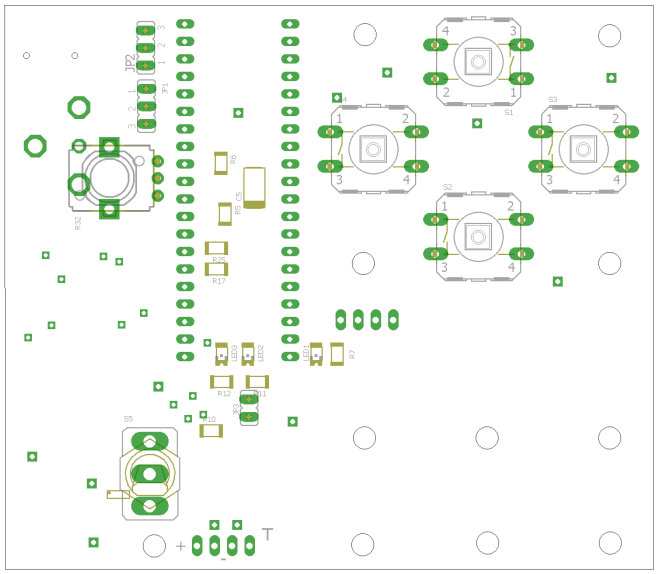
Obr. C.1: Návrh plošného spoja - pohľad z vrchu



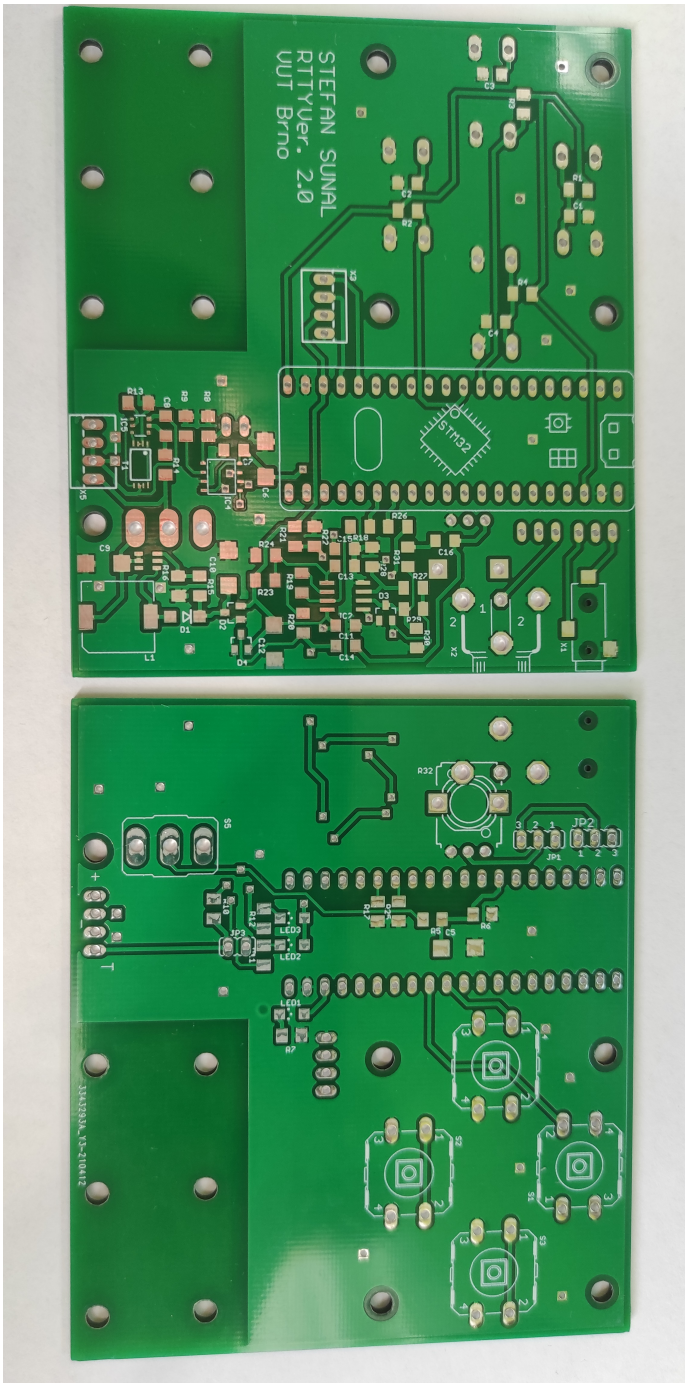
Obr. C.2: Návrh plošného spoja - pohľad zo spodu



Obr. C.3: Súčiastky na doske - pohľad z vrchu



Obr. C.4: Súčiastky na doske - pohľad zo spodu



Obr. C.5: Zrealizovaný plošný spoj

D Zariadenie

Obr. D.1: Realizácia zariadenia

E Obsah priloženého súboru zip

- `priloha` ..... koreňový adresár priloženého súboru
  - `DPS` ..... obázky plošného spoja a schémy
    - `DPS_BOTTOM.png`
    - `DPS_TOP.png`
    - `DPS_Foto.jpg`
    - `osadenie_Bottom.png`
    - `osadenie_TOP.png`
    - `schema.png`
  - `eagle` ..... súbory pre program Eagle
    - `RTTY.brd`
    - `RTTY.sch`
  - `Firmware` ..... Softvér pre mikrokontrolér
    - `Menu` ..... Knižnica zabezpečujúca UI
    - `RTTY` ..... Hlavný súbor
    - `RTTYDecoder` ..... Knižnica pre dekódovanie
  - `PC program`
    - `MainWindow.xaml.cs`
    - `MainWindow.xaml`
    - `RTTYapp.exe` ..... Spusitteľný súbor aplikácie